

2017/2018

RoboCupSoccer

Partie alimentation,
Microcontrôleur,
Communication Radio

RENAUD Pierre Louis

Table des matières

I.	Présentation du projet	2
II.	Cahier des charges	2
III.	Schéma fonctionnel	4
IV.	Recherche des solutions	7
V.	Analyse structurelle	13
VI.	Fabrication	22
VII.	Test et résolution des problèmes	28
VIII.	Etude logicielle.....	31
IX.	Conclusion.....	36
X.	Annexes.....	37

I. Présentation du projet

La RobotCup est une compétition internationale de robot footballeur. Il existe plusieurs catégories allant du robot bipède au robot holonome. Le but de la RobotCup sera de faire participer en 2050 l'équipe championne du monde de robot contre celle humaine.

Dans une moindre mesure, nous devrons participer à la RobotCup de 2020 dans la catégorie Robot Soccer et plus précisément la Small Size League (F180 League). Cette ligue impose de nombreuses règles pour y participer.

Le principe général est que nous devons concevoir et réaliser un robot de 25cm de diamètre sur 25cm de hauteur max avec un budget de 500€. Ce robot sera géré par un ordinateur à distance. Cet ordinateur aura accès à une caméra qui verra l'ensemble du terrain. Elle permettra au robot de voir où se situe la balle, ses coéquipiers ainsi que ses adversaires. Une intelligence artificielle sera installée sur l'ordinateur qui permettra un travail d'équipe entre robots.

II. Cahier des charges

Présentation du travail

Cahier des charges initial :

-Fonction FP1 : assurer l'alimentation du Robot ainsi que la recharge complète de la batterie, qui permettra une autonomie comprise entre 20 min et 35 min.

Le système doit permettre le rechargement complet de l'alimentation. Elle fournit tous les courants et tensions nécessaires à l'ensemble du robot.

-Fonction FP2 : assurer la communication sans fil.

Le système doit permettre de connaître l'état du robot et également pouvoir transmettre des ordres envoyés par l'utilisateur.

-Fonction FP3 : assurer les gestions des différentes entrées/sorties du robot, des appels de fonctions, protocoles de communication, ainsi que l'ordonnancement des fonctions du robot.

Cahier des charges approfondi :

Fonction FP1 : ALIMENTATION

1. -Doit satisfaire au besoin énergétique de tout robot.
2. -Doit avoir une autonomie comprise en 20 min et 35min.
3. -Doit permettre un rechargement complet et une gestion du niveau de la batterie.
4. -Doit permettre de connaître l'état de charge.
5. -Doit avoir une protection du système.
6. -Doit prendre en compte les contraintes d'espace des systèmes embarqués.

Fonction	Critère	Valeur
FP1 : alimentation électrique du robot.	Fournir les tensions. Fournir le courant.	7.5V, 3.3V et une tension de batterie comprise entre 12.6V et 9V (alimentation moteur)
FS1 : autonomie	Doit posséder une batterie LiPo 3S.	20 min a 35 min
FS2 : rechargement	Un rechargement interne.	Un système permettant de charger et de gérer le cycle de charge de la batterie doit être prévu.
FS3 : IHM	IHM	Indication de l'état de batterie.
FS4 : Protection du système	Doit protéger le robot.	Des diodes de non-retour et des fusibles seront prévus pour sécuriser le robot.
FS5 : contraintes de l'embarqué	Taille, forme	La carte ne devra pas dépasser un diamètre et une hauteur de 25cm.Elle doit pouvoir être placée dans une forme cylindrique.

Fonction FP2 : Communication

1. -Doit pouvoir communiquer entre le robot et un ordinateur.
2. -Doit avoir une interface sur l'ordinateur pour connaître l'état du robot.

Fonction	Critère	Valeur
FP1 : Communiquer	Un module radio imposé sera utilisé.	Module nRF24L01
FS1 : Restitution de l'information	Une interface utilisateur pour transmettre les instructions au robot devra être mise en place	Utilisation de PUTTY avec une communication série

Fonction FP3 : Gérer

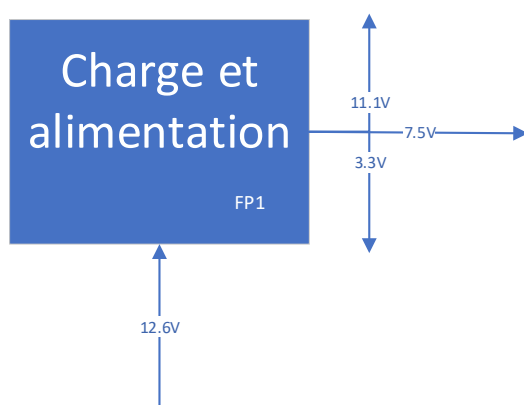
1. -Doit pouvoir gérer des entrées/sorties demandées par les différentes parties.
2. -Doit pouvoir assurer les communications entre les différentes parties.
3. -Doit pouvoir assurer les calculs demandés.
4. -Doit pouvoir assurer l'ordonnancement des différents programmes.
5. -Doit prendre en compte les contraintes d'espace des systèmes embarqués.

Fonction	Critère	Valeur
FP1 : assurer les entrées/sorties	Doit avoir le nombre suffisant d'entrées /sorties, de protocole de communication.	Doit avoir des ports GPIO. Avoir les protocoles de communication suivant : SPI, I2C.
FS1 : assurer l'ordonnancement	Doit permettre une gestion des fonctions utilisées par le robot.	Aucune demande spécifique n'est stipulée pour cette fonction.

III. Schéma fonctionnel

Schéma fonctionnel de niveau 1

Partie alimentation

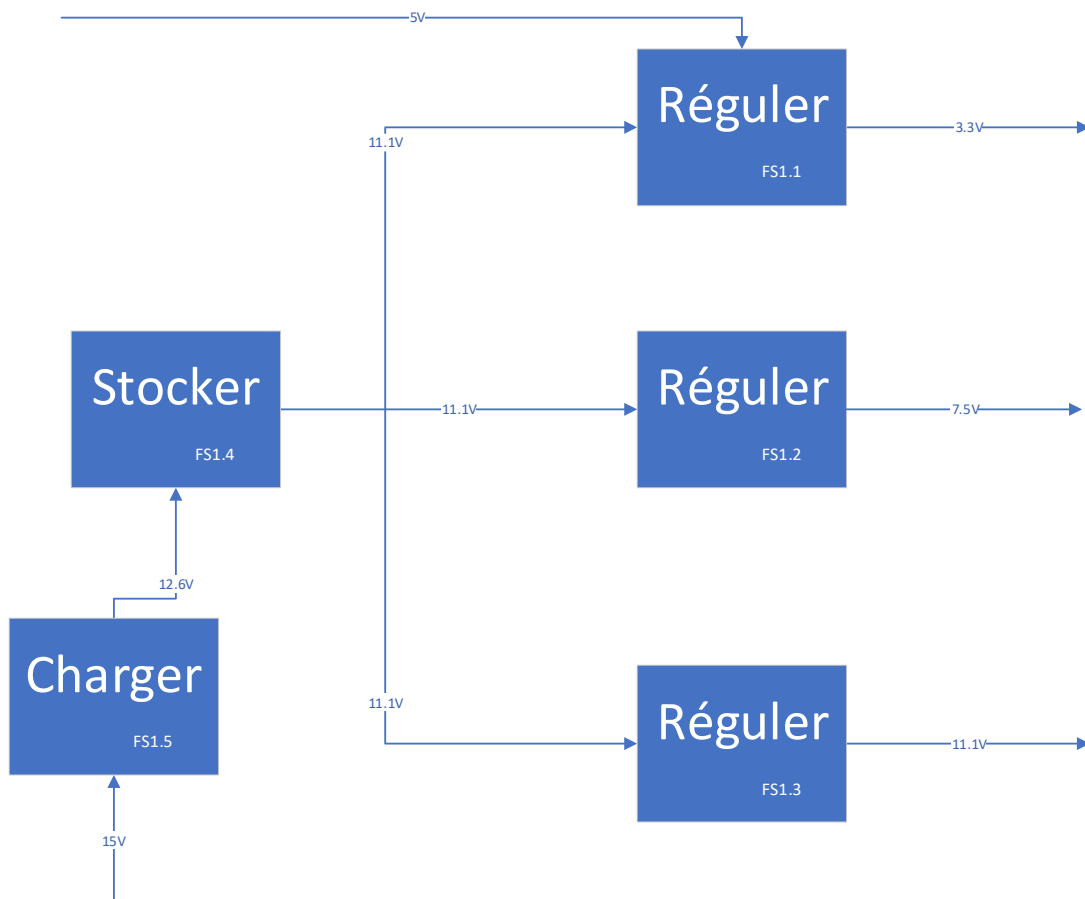


Partie microcontrôleur



Schéma fonctionnel de niveau 2

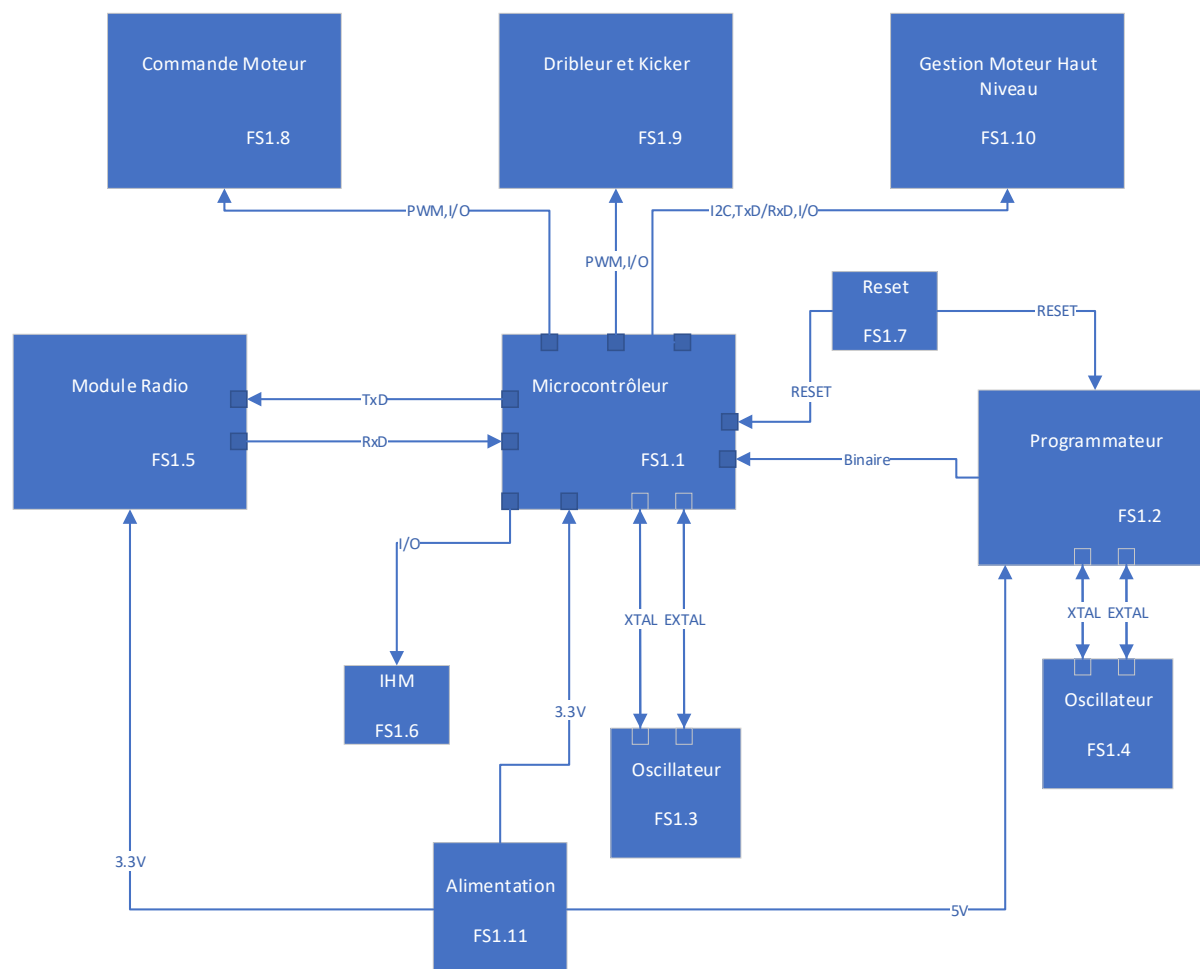
Partie alimentation



Description des signaux

Signaux	Forme	Commentaire
15V	Continue	Tension d'alimentation du chargeur
12.6V	Continue	Tension de charge de la batterie
11.1V	Continue	Tension d'entrée de chaque régulateur
3.3V	Continue	Tension d'alimentation de la partie microcontrôleur
7.5V	Continue	Tension d'alimentation du dribbler et du kicker
11.1V	Continue	Tension d'alimentation des moteurs directement branchés sur la batterie
5V	Continue	Sortie d'alimentation USB connectée sur le régulateur 3.3V

Partie microcontrôleur



Description des signaux

Signaux	Forme	Commentaire
TxD	Signal Carré	Signaux de la transmission de la SPI
RxD	Signal Carré	Signaux de la réception la SPI
I/O	Signal Carré	GPIO
XTAL	Signal Sinusoïdal	Signal de l'oscillateur
EXTAL	Signal Sinusoïdal	Signal de l'oscillateur
RESET	Signal Carré	Permet de RESET le microcontrôleur
BINAIRE	Signal Carré	Permet d'envoyer le code d'instruction au microcontrôleur
PWM	Signal Carré	Signal PWM
I2C	Signal Carré	Signal I2C permettant de communiquer avec la gestion des moteurs
3.3V	Continue	Alimentation du microcontrôleur et du module radio
5V	Continue	Alimentation du programmeur

IV. Recherche des solutions

Partie alimentation

Pour satisfaire le cahier des charges, des solutions ont dues être recherchées par rapport aux demandes des différentes partie :

Nom	Tension (V)	Courant (A)	Puissance (W)
Vincent	7,5	4	30
Clement M	12	8,08	96,96
Clement C	3,3	0,1739	0,57387
Pierre Louis R	3,3	0,163	0,5379
	Total courant (A) :		12,4169
	Total Puis(W) :		128,07177

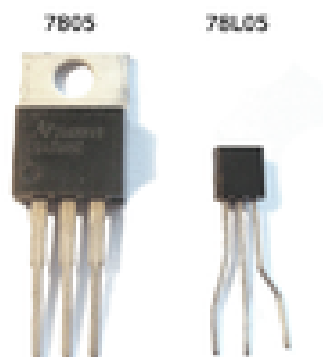
Tableau récapitulatif des besoins en Tension et Courant

Partie Alimentation

Pour alimenter le robot, l'utilisation de régulateurs est nécessaire pour avoir les tensions demandées par le cahier des charges. Pour cela il existe deux technologies de régulateur : les régulateurs linéaires et les régulateurs à découpage. Une étude technique sur chacun d'entre eux a été effectuée.

Régulateur linéaire (LDO) :

Ce composant permet d'abaisser et de stabiliser une tension qu'on fournit. Il existe soit une version a tension de sortie fixe ou bien en version ajustable. Il fonctionne comme une résistance variable.



Source : Wikipédia

<u>Avantage</u>	<u>Inconvénient</u>
Facile à mettre en œuvre	Rendement médiocre
Faible coût	Grande puissance dissipée
Faible bruit	Utilisation d'un dissipateur
	Encombrement plus important

Régulateur à découpage :

On peut voir ci-contre un schéma de ce régulateur.

Ce système permettrait d'avoir un gain de place sur le montage. Il en existe plusieurs types dont les principaux sont : Buck et Boost.

Le convertisseur Buck ou hacheur série est un régulateur qui abaisse une tension d'entrée en tension de sortie plus faible.

Le principe est de faire varier le rapport cyclique.

Dans le cas du Buck on a :

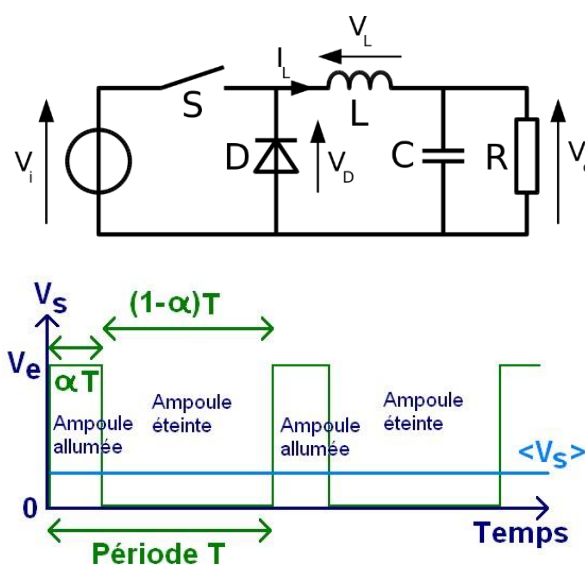
$$V_s = \alpha V_e$$

Avec

V_s : la tension de sortie en V

V_e : la tension d'entrée en V

α : le rapport cyclique



Source Wikipédia et astuce-pratique

<u>Avantage</u>	<u>Inconvénient</u>
Rendement élevé	Coût un peu plus élevé
Faible dissipation thermique	Bruit parasite important
Faible encombrement	

En conclusion pour le choix des régulateurs : 3 sont à découpage et un sera linéaire

Régulateur linéaire :

Un régulateur linéaire est utilisé pour la charge de la batterie il s'agit du LM217T. Il peut fournir 2.2A et possède un rendement de 84%. La tâche de ce composant est juste de fournir une tension fixe. Il sera connecté à adaptateur secteur 220V/15V pour la recharge de la batterie. Le régulateur linéaire permet de simplifier la carte. Sa mise en œuvre simple et son faible coût permettra d'acheter des composants plus élaborés pour la suite du projet.

Régulateurs à découpage :

-Pour la tension de 7.5V : LM2677S-ADJ/NOPB. C'est un régulateur ajustable qui peut fournir un courant allant jusqu'à 5 ampères qui à alimenter le dribbler et le kicker. Il possède un rendement compris en 84% et 92%, cette valeur n'étant pas très précise sur la datasheet.

-Pour la tension de 3.3V : deux régulateurs ont été choisis : un connecté à l'entrée de la batterie et l'autre connecté aux ports USB. Celui qui est connecté à la batterie et le 700-MAX5033AASAT, c'est un régulateur à tension de sortie fixe de 3.3V. Il peut fournir un courant de 500mA. Il possède un rendement de 94%. Celui connecté au port USB est un LM3671MF-1.5/NOPB. Il s'agit également d'un régulateur 3.3V à tension fixe qui peut délivrer un courant de 600mA et possède un rendement de 95%.

Ces régulateurs à découpage sont choisis essentiellement pour leurs meilleurs rendements. Etant sur alimentation batterie, ils permettront d'optimiser l'autonomie de celle-ci.

Les moteurs du robot seront directement connectés à la batterie.

Récapitulatifs :

Pour l'alimentation 7.5V :

Régulateur à découpage LM2677S-ADJ/NOPB : 7.5V, 5A avec un rendement compris entre 84% et 92%.

Pour l'alimentation 3.3V :

Deux régulateurs à découpage : 700-MAX5033AASAT 3.3V 500mA avec un rendement de 94% et le LM3671MF-1.5/NOPB 3.3V 600mA avec un rendement de 95%.

-Avec les régulateurs et les transistors Mosfet un autre problème est apparu il s'agit de la chauffe de ces derniers. De ce fait, une étude de dissipation thermique a été effectuée.

Calcul de la dissipation thermique et des besoins du radiateur :

Puissance dans le régulateur linéaire :
 $Pd = (Vin - Vout) * Iout$

Puissance des régulateurs à découpage :
 $Pd = \frac{Vout * Iout}{\eta} - Vout * Iout$

Puissance des transistor Mosfet :

$$Pd = Vds * Id + RDson * Idef^2$$

Calcul pour savoir s'il faut un dissipateur :

Il faut comparer la température de jonction max donné par la datasheet du composant.

Puis calculer la température par rapport à celle donnée :

$$T \text{ sans radiateur} = Rthj - a(^{\circ}C/w) * Puissance \text{ Max}(W) + T \text{ Ambiante}(^{\circ}C)$$

Avec :

-V en Volt

-I en ampère

Choix du radiateur :

$$Rthr - a = \frac{Tj \text{ max} - T \text{ ambiante Max}}{Pd \text{ max} - (Rthj - b) - (Rthb - r)}$$

Pour les transistors Mosfet deux radiateurs d'une température de jonction 8.6 °C/W et pour le régulateur à découpage un radiateur de 11 °C/W.

Chargeur

Pour la charge de la batterie, le plus compliqué est de surveiller le cycle de charge et de décharge de la batterie. Avant de penser à mettre en œuvre un composant spécialisé, plusieurs montages à base d'AOP et de transistor ont été pensés. Mais ces derniers apportent moins de fonctionnalités par rapport à un CI spécialisé. C'est pourquoi un CI spécialisé a été choisi apportant plus de fonctionnalité comme la surveillance des sous charges, des surcharges et une fonction d'équilibrage des cellules. Plusieurs types de CI existent mais pour avoir de meilleures performances, il est préférable de prendre un composant spécialisé pour la technologie de batterie choisies. Après des recherches un circuit s'est démarqué des autres, il s'agit du S-8254AAWFT-TB-G un CI de gestion et de protection des batteries : Li-Po et Li-ion. Avec l'ensemble de composants qui l'entoure, il permet de gérer la charge et la décharge de la batterie. Suivant la référence du composant choisie il aura des tensions de coupure automatique en cas de surcharge ou sous charges pour la protection de la batterie.

Batterie

Dans notre cas la technologie et le nombre de cellule ont été imposés. Il s'agit d'une technologie Li-Po à 3 cellules.

La technologie Li-Po a l'avantage d'être légère et bien adaptée au modélisme. Néanmoins elle nécessite une surveillance rigoureuse lors de sa charge et décharge. Nous devons calculer sa capacité pour pouvoir satisfaire le temps demande*é.

Nom	Tension (V)	Courant (A)	Puissance (W)
Vincent	7,5	4	30
Clément M	12	8,08	96,96
Clément C	3,3	0,1739	0,57387
Pierre Louis R	3,3	0,163	0,5379
Total courant (A) :		12,4169	
Total Puis(W) :			128,07177
Autonomie en heure	0,58	0,41	0,25
Energie Conso Wh	74,2816266	52,5094257	32,0179425
Tension batterie V	11,7	11,7	12,6
Ah	6,348856974	4,487985103	2,541106548
Marge d'erreur	1,2	1,2	1,2
Ah (avec erreur)	7,618628369	5,385582123	3,049327857

Avec ces différents calculs une batterie de 5000mAh est choisie.

Partie transmission sans-fil

Pour gérer la transmission sans fil un module radio nous a été imposé. Il s'agit du nRF24L01 un module émission/réception. Il fonctionne sur la gamme de fréquence des 2.4Ghz.

Partie microcontrôleur

Pour cette partie un microcontrôleur de famille NXP avec une architecture CISC nous est imposé. Le choix du microcontrôleur se fera en rapport aux besoins des différentes parties du robot. Cette gamme de microcontrôleur propose différents types de boitiers qui déterminent le nombre de broche qui peut aller de 28 à 100.

Clément C			Clement M			Vincent		
	I2C	PORT		PWM	PORT		PWM	
	SDA_2	3		TPM3CH0	34		TPM3CH4	78
	SCL_2	4		TPM3CH1	33		TPM3CH5	77
				TPM3CH2	25			
	GPIO	42		TPM3CH3	24		GPIO	PORT
		41					PTDD_3	57
		40		GPIO	74		PTDD_4	56
		39			73		CAN	53
		23			72		CAN	49
		22			71	Pierre Louis		
		21			68		CAN	52
		20			67	SPI	MOSI	75
	SPI	PORT			66		SPSCK1	70
	Mosi	1			65		MISO	76
	Miso	58					SS1	69
	Clk	2					KBI1P1	62
						GPIO		
							26	PTDD_7
							27	PTDD_6
							28	PTDD_5
							61	CE_μ

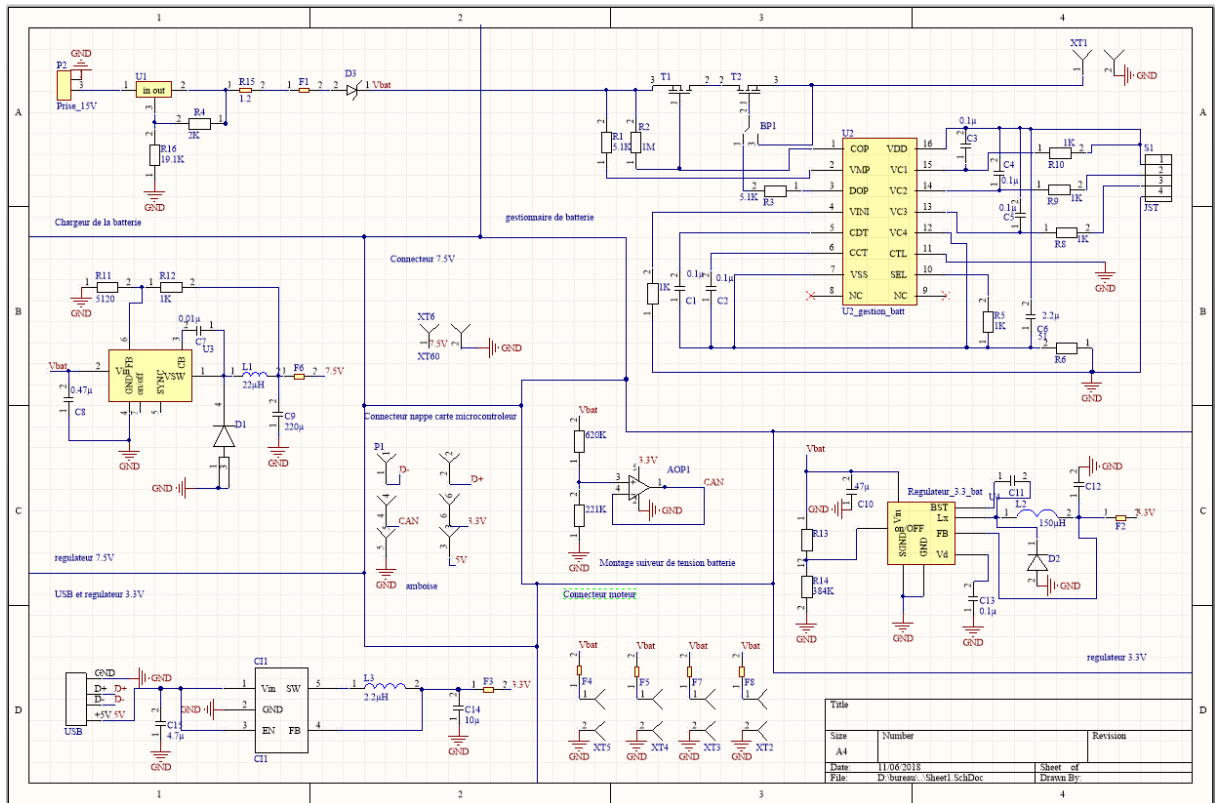
Tableau récapitulatif des besoins des différentes parties au niveau du microcontrôleur

Avec les différentes demandes des collègues, un microcontrôleur 80 broches est choisi. Il s'agit du MC9S08QE128CLK. Il possède 8 KB de RAM et 128KB.

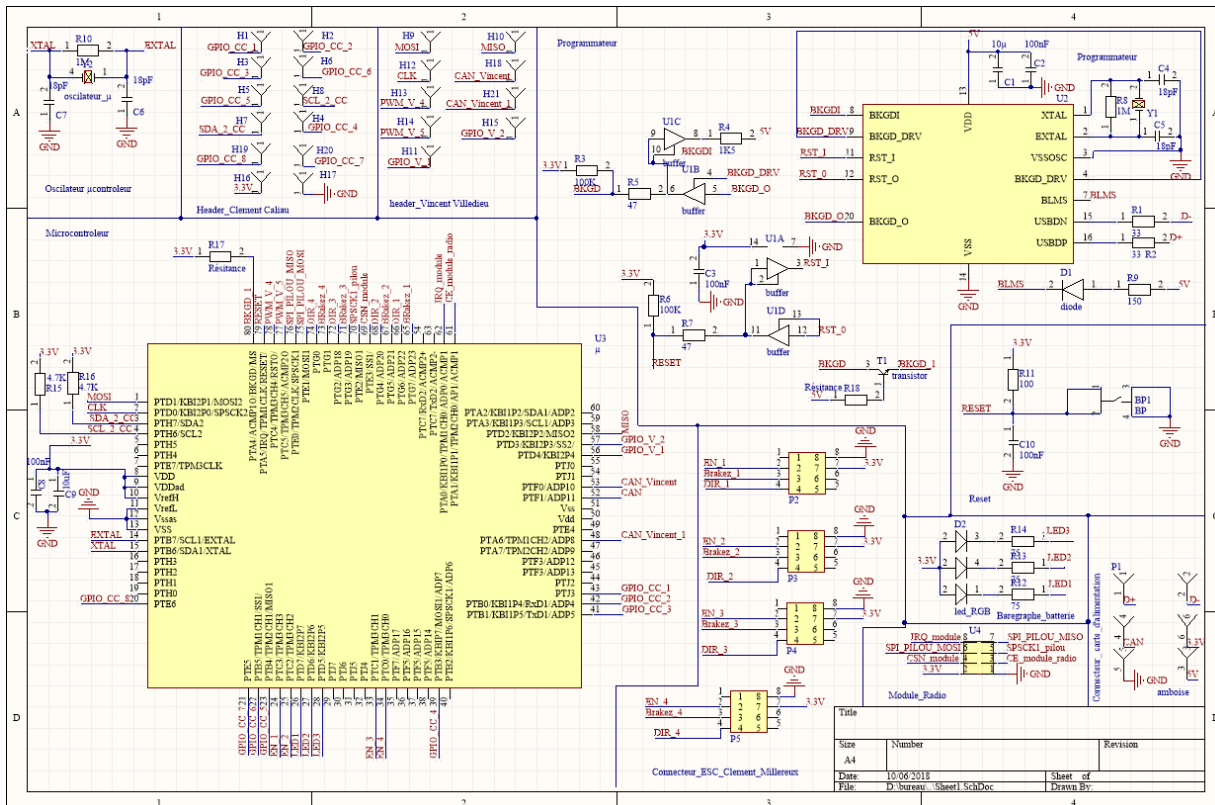
V. Analyse structurelle

Schéma structurelle

Partie alimentation



Partie microcontrôleur



Chargeur de batterie :

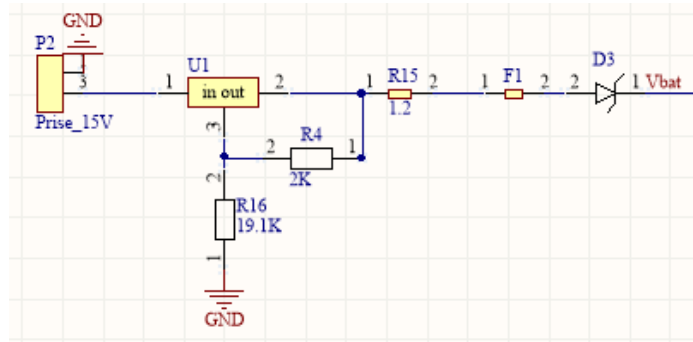
Le chargeur de batterie est basé sur un LM217. Il s'agit d'un régulateur de tension linéaire réglable. Pour calculer sa tension de sortie on utilise la formule suivante :

$$VS = V_{ref} \left(1 + \frac{R2}{R1}\right)$$

Avec $V_{ref} = 1.25V$

Dans notre cas la tension de sortie est de 14 V. On choisit une tension plus haute pour compenser les pertes de la diode et des transistors qui seront expliquées plus bas.

Pour éviter que la batterie vienne se décharger dans le régulateur. Une diode Schottky est placée à la sortie du régulateur. On choisit ce type de diode car sa chute de tension a ses bornes est plus faible. Egalement pour la sécurité de la batterie et du système un fusible de 1.5A est placé en sortie du régulateur.



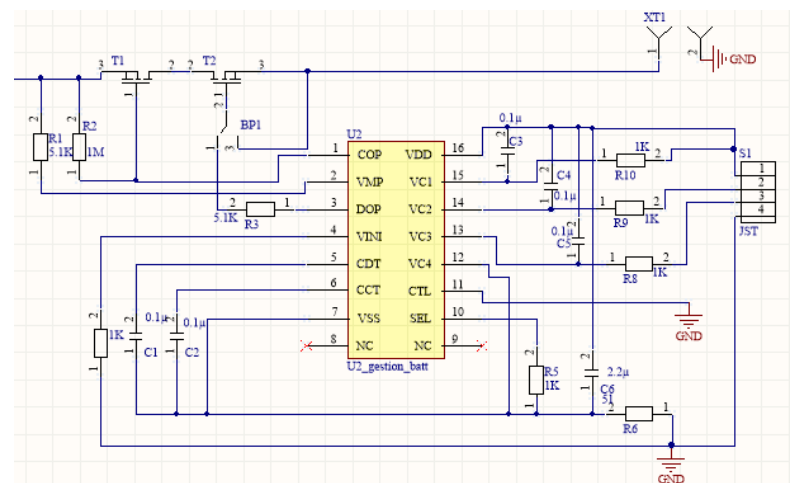
Description des broches

- 1 : Vin
- 2 : Vout
- 3 : adjust permet de régler la tension de sortie.

Gestionnaire de batterie :

Afin de gérer les différents cycles de la batterie, le composant S-8254AAWFT-TB-G est choisi. Il permet de gérer la charge et décharge de la batterie. Il se connecte au port d'équilibrage directement intégré sur la batterie. Il surveille les différentes cellules grâce à ce port.

On lui connecte également aux broches 1(COP) et 3 (DOP) deux transistors. Le premier permet de la couper ou la connecter et il s'occupe de la charge de la batterie. Le second lui s'occupe de la décharge en se coupant si la batterie passe sous un seuil. Les différents composant ainsi que leurs valeurs qui entoure le CI sont fournis par la datasheet.



Description des broches :

1 : contrôle le transistor de charge	9 : NC
2 : permet la détection de la tension de la batterie	10 : permet de sélectionner le nombre de cellules.
3 : contrôle le transistor de décharges	11 : contrôle manuel des transistors.
4 : broche de surveillance de surintensité.	12, 13, 14, 15 : broche de connexion des cellules de la batterie.
5 : permet de régler le délai de détection de surintensité.	16 : broche d'alimentation du CI.
6 : permet de retarder le courant de surcharge.	
7 : permet de brancher une alimentation négative.	
8 : NC	

Régulateur 7.5V :

Pour alimenter le dribbler et le kicker une alimentation leur est dédiée.

Il s'agit du composant *LM2677S-ADJ/NOPB*. C'est un régulateur à sorties ajustables. Il fournit un courant de 4.5A. Il prend en entrée la tension de la batterie pour sortir du 7.5V.

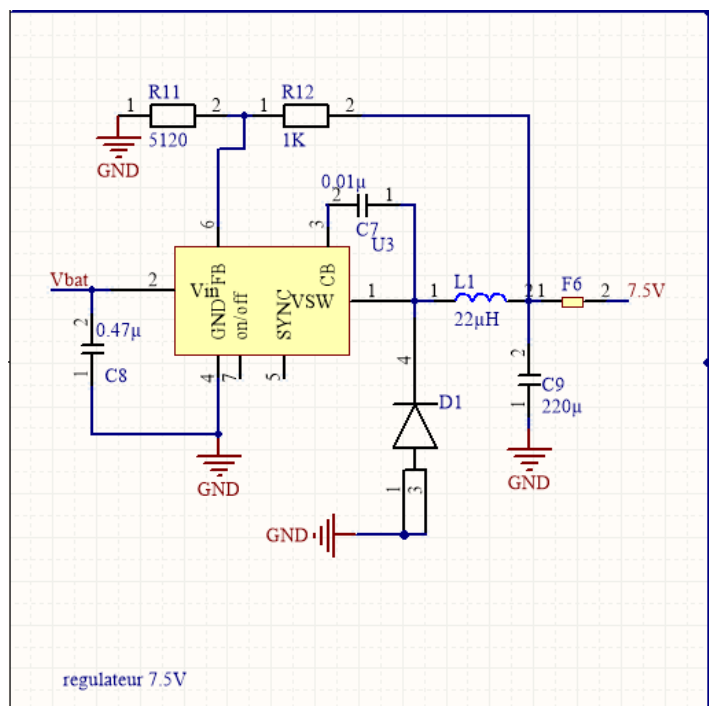
Avec la résistance R11 et R12 on peut régler la tension de sortie grâce au calcul suivant :

$$V_{out} = V_{fb} \left(1 + \frac{R_{12}}{R_{11}} \right)$$

Avec $V_{fb} = 1.21V$

La self connectée en sortie est préconisée par la datasheet parmi un tableau de valeurs. Elle est choisie par rapport à la tension de sortie désirée.

La diode Schottky D1 est préconisée par la datasheet. Elle est dimensionnée par rapport aux besoins de courant et devra fournir le régulateur.





Alimentation 3.3V USB :

Pour pouvoir programmer le microcontrôleur sans devoir toujours se connecter à la batterie, un régulateur de tension est branché au port USB. Ce composant est choisi pour sa mise en œuvre très simple.

La self L3 est donné par la datasheet.

Les condensateur C15 et C14 sont calculés grâce au calcul de l'impédance du condensateur qui va court circuiter les hautes fréquences vers la masse.

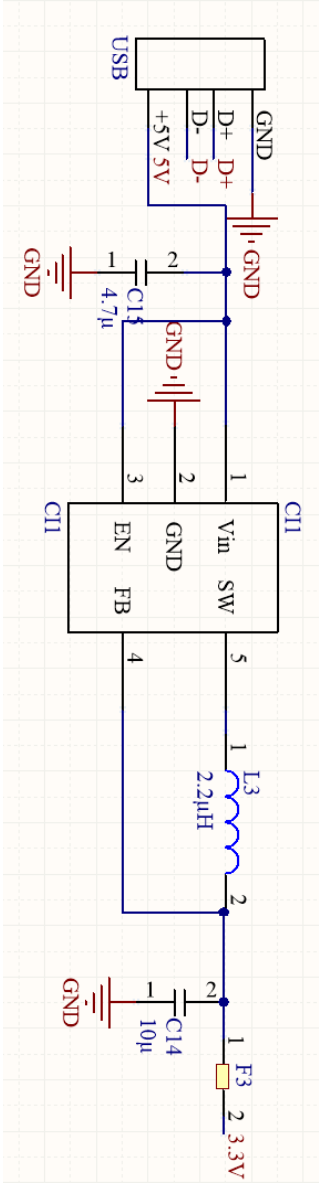
$$Z_c = \frac{1}{C2\pi f}$$

Avec
-f en hertz
-C en farad

Un fusible est également mis en place pour protéger le reste du système.

Description des broches

- Vin (01) : broche d'entrée du régulateur.
- GND (2) : connexion du CI a la masse
- EN (3) : permet l'activation ou la désactivation du CI si EN>1V alors active le CI, si EN<0.4V désactive le CI.
- FB (4) : Dans notre cas cette broche est directement reliée à la borne du condensateur. Elle sert à ajuster la tension de sortie dans les régulateurs ajustables.
- SW (5) : La tension de sortie est égale à la tension moyenne à la broche SW dans notre cas à +3.3V.



Montage suiveur de tension :

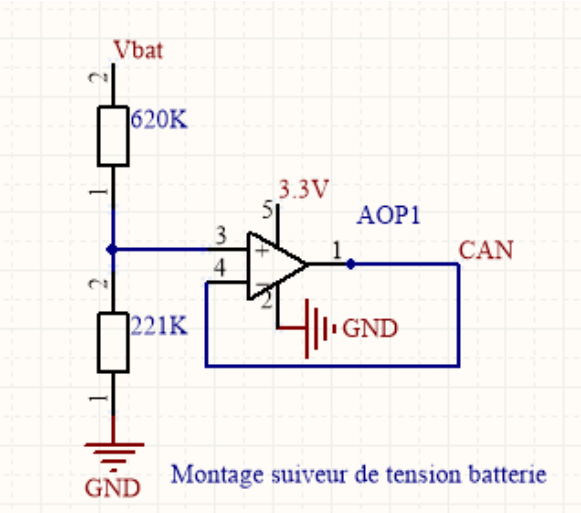
Pour connaitre la tension de la batterie, on utilise une ADC intégrée dans le microcontrôleur.

Ce montage permet d'adapter la tension de la batterie, vers l'ADC du microcontrôleur.

$$Vs = Vbat \frac{221K}{620K + 221K}$$

$$Vs = Vbat * 0.26$$

Vbat(V)	12.6	11.1	9
Vsat(V)	3.27	2.88	2.34

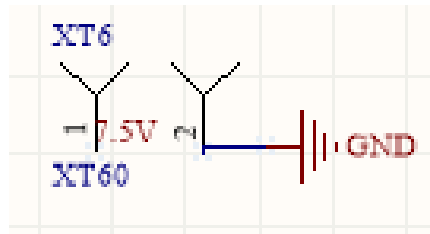
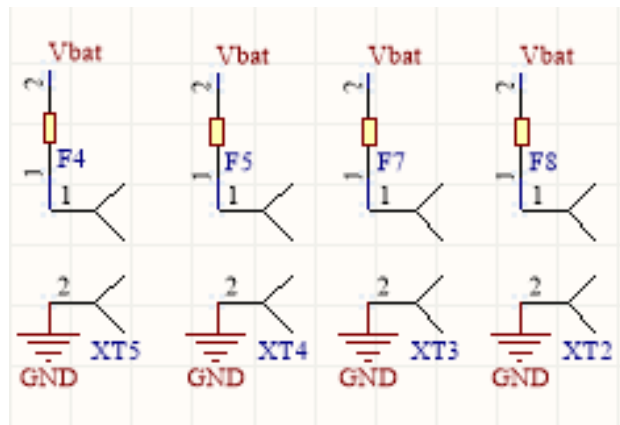


Connecteur moteur :

Pour connecter les moteurs à la carte d'alimentation on utilise des connecteurs XT60. Ils permettent de faire passer un grand courant jusqu'à 60 ampères. Ils sont directement connectés à la batterie. Des fusibles de protection de 10A à action lente sont disposés à chaque connecteur. Ils permettent de protéger la batterie en cas de gros pic de courant et également de protéger les moteurs en cas de défauts dans la batterie.

Connecteur kicker et dribbleur :

Un connecteur XT60 est utilisé pour brancher le kicker et dribbleur. Le connecteur étant surdimensionné par rapport à la consommation de courant demandée pour la partie. Ces derniers étant vendus en pack de 5, une économie a été faite en évitant d'acheter un autre connecteur.



Partie microcontrôleur

Microcontrôleur :

Un microcontrôleur 80 broches est utilisé pour faire fonctionner le robot.

Les résistances R15 et R16 sont deux résistances de tirage (pull up) pour le protocole I²C.

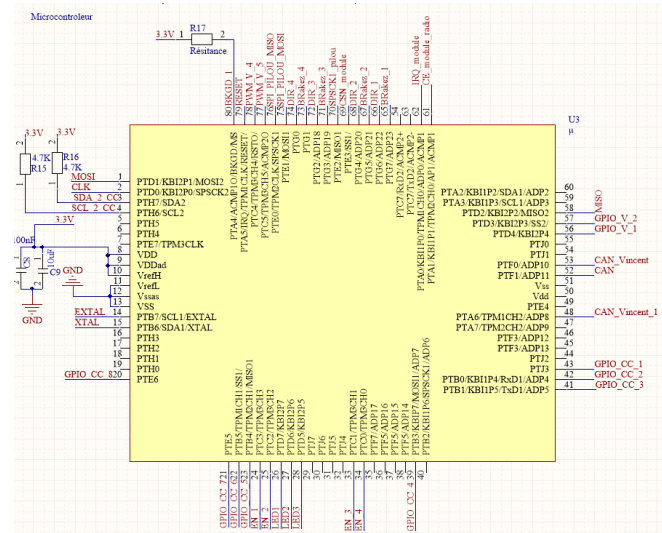
Les condensateurs C8 et C9 sont deux condensateurs de découplage.

$$Z_c = \frac{1}{C2\pi f}$$

Avec

-f en hertz

-C en farad



Programmateur :

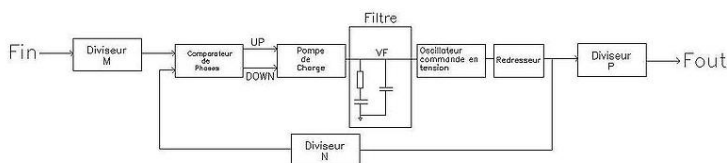
On utilise un MC9S08JS16CWJ qui possède 512O de ram et fonctionne à un fréquence max de 24MHz. Ce dernier sera flashé pour programmer le microcontrôleur qui lui a été associé. Grâce au buffer nommé U1 il permettra d'adapter la tension BKGD du programmeur à celle du microcontrôleur. En effet le programmeur est alimenté en 5V du coup ses signaux de sortie sont eux aussi à 5V. Alors que le microcontrôleur est alimenté en 3.3V une adaptation de tension est donc nécessaire pour pas l'endommager.

Egalement un transistor T1 est positionné entre les broches du BKGD du micro et du programmeur. Lorsque le programmeur n'est pas alimenté la broche BKGD est à l'état haut ce qui empêche le bon fonctionnement du microcontrôleur. De ce fait, la base du transistor est branchée au 5V et dès que celui-ci est branché, le transistor devient passant et permet de programmer le microcontrôleur.

Oscillateur microcontrôleur :

On utilise un quartz de 32.768Khz avec ses composants une résistance d'1 M Ohm et de deux condensateurs de 18pF.

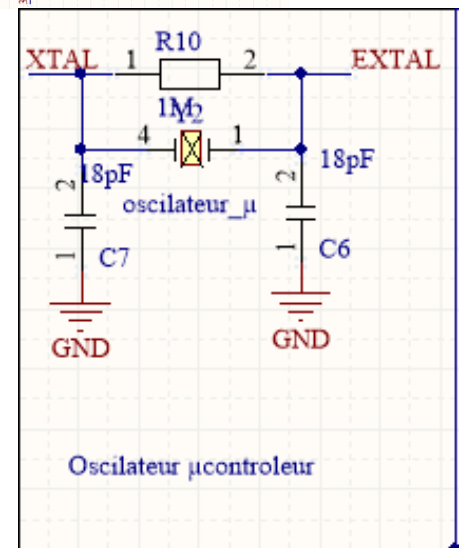
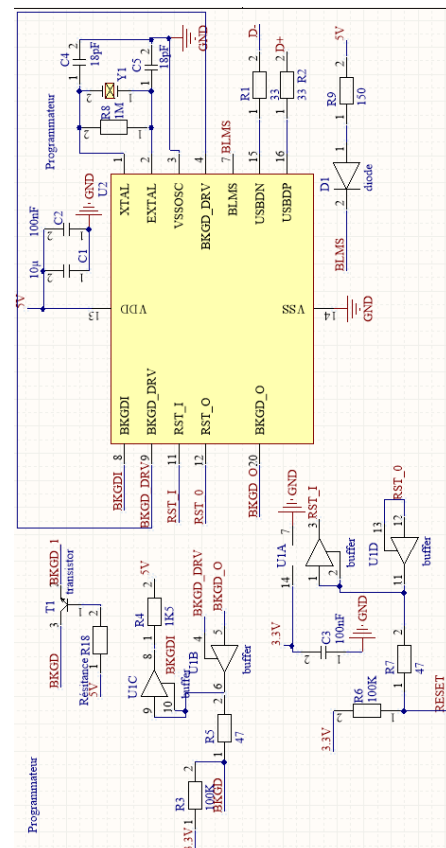
On l'associe à la PLL (Phase-Locked Loop) du microcontrôleur pour avoir une fréquence de 50.33kHz. Une PLL est un système qui permet d'asservir la fréquence et ainsi avoir une fréquence plus élevée.



$$F_{out} = F_{in} * N / (MP)$$

Schéma du système qui se trouve à l'intérieur du microcontrôleur ainsi que la formule générale de la fréquence de sortie.

On utilise la fonction qui permet de paramétrer le coté hardware du microcontrôleur.



Component Inspector - Cpu

Basic Advanced

Properties Methods Events Build options Resources

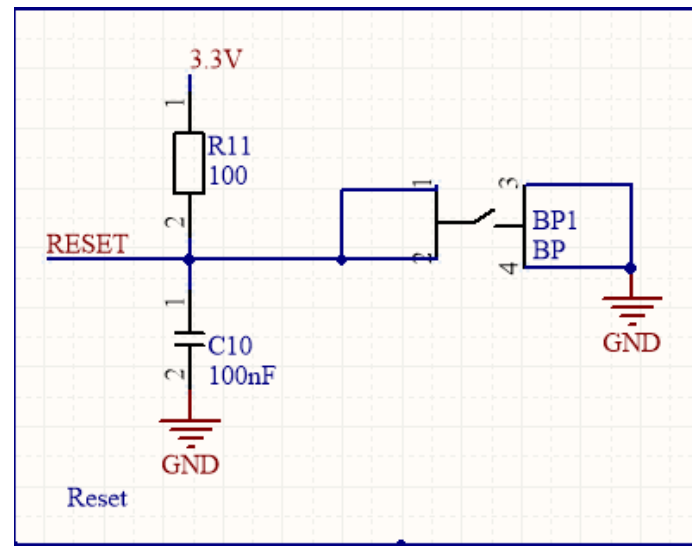
Name	Value	Details
CPU type	MC9S08QE128CLK	
✓ Clock settings		
✓ Internal clock		
Internal oscillator frequency	32.768	32.768 kHz
✓ External clock	Enabled	
✓ Clock source	External oscillator	
Clock frequency [MHz]	0.032768	0.032768 MHz
Clock range	Low frequency	<31.25 kHz, 1 MHz>
> Low-power modes settings		
Initialization interrupt priority	interrupts enabled	1
> CPU interrupts		
✓ Enabled speed modes		
✓ High speed mode	Enabled	
High speed clock	External Clock	32.768 kHz
Internal bus clock	25.165824	25.165824 MHz (50.331648/1/2)
Fixed frequency clock [MHz]	0.016384	
✓ FLL mode	Engaged	FEE
✓ Ref. clock source	External Clock	
Ref. clock source frequency [MHz]	0.032768	0.032768 MHz
Ref. clock frequency [MHz]	0.032768	0.032768 MHz (0.032768/1)
DCO mode	Auto select	Default (1536)

Reset

Pour redémarrer le microcontrôleur on utilise la fonction RESET du micro associé au montage ci-contre. Il faut que le potentiel entre la résistance et le condensateur soit à l'état bas pendant 100ns.

$$\tau = R11 * C10$$

$$100nS = 100 * 100nF$$



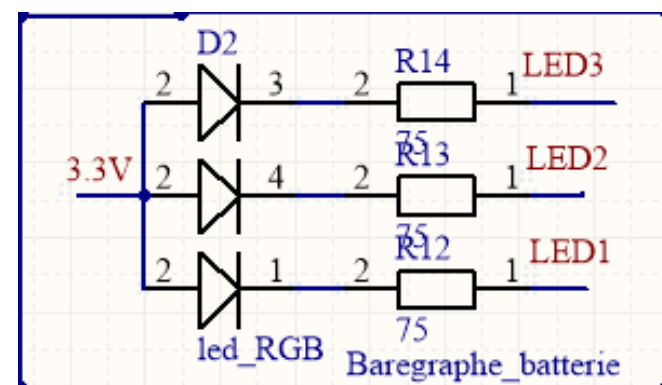
Bargraphe led :

Il s'agit de trois led connectées en cathode commune. Elle permet d'avoir un visuel sur la quantité de batterie restante. Il y a trois couleurs ; rouge, orange et vert. Etant connectée en anode commune il faut mettre la broche du microcontrôleur à l'état bas pour activer la led.

Les résistances sont de 75 ohm chacune.

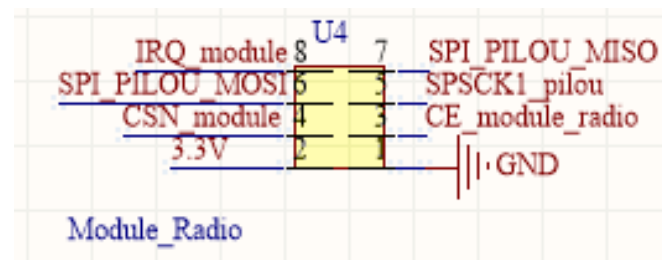
On obtient leurs valeurs de la manière suivante :

$$R = \frac{V_{cc} - V_{led}}{I_{led}}$$



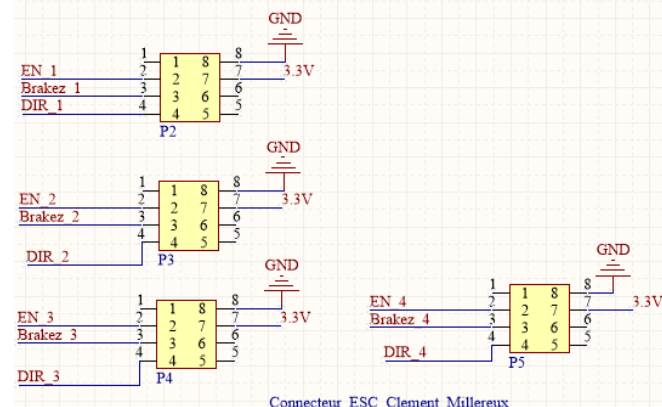
Connecteur module radio :

Il s'agit du connecteur qui permet de relier les différentes broches du module au microcontrôleur.



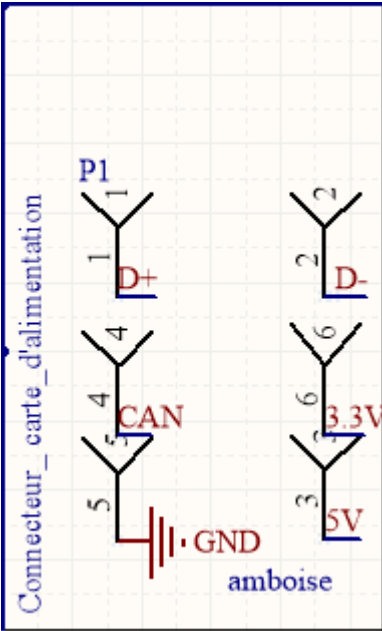
Connecteur ESC :

Pour envoyer les commandes aux moteurs on utilise des connecteurs JST 8 broches. Ils sont directement connectés au microcontrôleur.



Connecteur de liaison carte alim carte microcontrôleur :

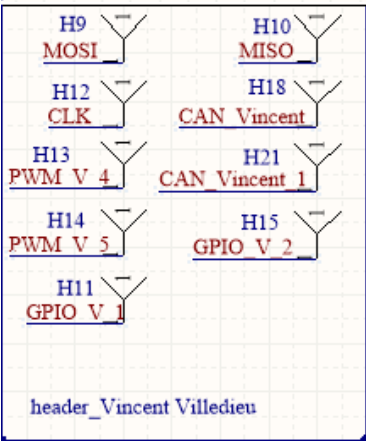
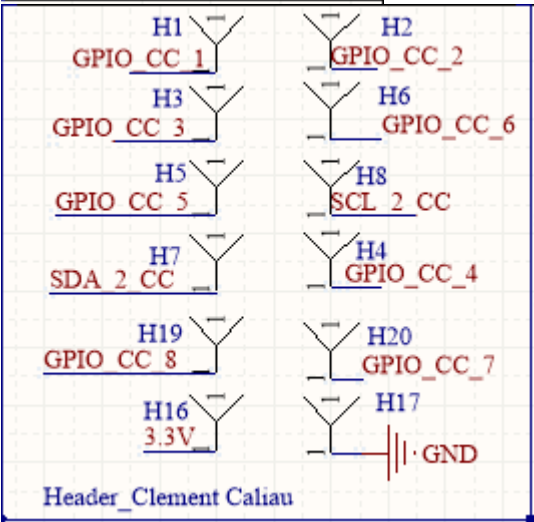
Pour relier les tensions d'alimentation et les données des USB on utilise un JST 6 broches.



Header

Il s'agit de header pour connecter les GPIO et les protocoles de communication avec la partie commande moteur haut niveau.

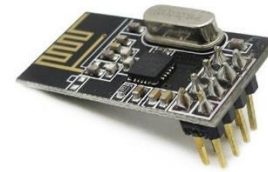
Egalement les headers de la commande des kicker et dribbleur.



Partie module radio

-il s'agit d'un module radio d'émission/ réception. Il émet sur la bande des 2.4Ghz en utilisant une modulation GFSK (Gaussian Frequency Shift Keying modulation). Il s'agit d'une modulation par déplacement de fréquence mais sur celle-ci on rajoute un filtre Gaussien. Ce système limite le champ spectral du signal.

Sa consommation est également faible (11.3mA en Tx et 12.3mA Rx) ce qui en fait un composant idéal pour un système embarqué. On l'alimente entre 1.9V et 3.6V.



VI. Fabrication

Nomenclature

Nomenclature alimentation

Nom du produit	Nom du fabricant	Référence fabricant	Nom du fournisseur	Référence fournisseur	Quantité	Prénom	état commande	Prix unitaire HT	Total HT	Prix TTC
Condensateur 220µ	Mu rata	GRM31CR60G227ME11L	Farnell	2611950	5	Renaud	Requ	0,178333333	1,78333333	2,14
Régulateur Buck 7,5V	TI	LM2677S-ADJ/NOPB	Farnell	1564714	1	Renaud	Requ	0,35	1,75	2,1
Cavalliers	Pro-signal	PSG-JRBN40-FF	Farnell	2452751	1	Renaud	Requ	4,841666667	4,84166667	5,81
BP, spdt	Multicomp	1MS1T1B5M1QE	Farnell	9473378	3	Renaud	Requ	1,416666667	1,41666667	1,7
Condensateur 47µ	Murata	GRM31CE70J476ME15L	Farnell	1907522	5	Renaud	Requ	0,250833333	1,25416667	1,505
Fusible 4,5	Littelfuse	1206SF5450F/32-2	Farnell	1843834	5	Renaud	Requ	0,214166667	1,07083333	1,285
Programmeur	NXF	MCS508J16CWJ	Farnell	1704517	1	Renaud	Requ	2,816666667	2,81666667	3,38
Dissipateur thermique reg 7,5V		7109DG	Farnell	1651790	1	Renaud	Requ	0,0975	0,975	1,17
Dissipateur thermique Mosfet		SW50-4G	Farnell	1213442	1	Renaud	Requ	0,0975	0,975	1,17
Diode Schottky 7,5	ON	MBRD835LT4G	Farnell	1611349	5	Renaud	Requ	0,155	0,775	0,93
Resistance 221K_AOP	Panasonic	ERJP08F2213V	Farnell	2327106	10	Renaud	Requ	0,7225	0,7225	0,867
Fusible 10A	Schurter	3413.0328.22	Farnell	1829471	5	Renaud	Requ	0,675833333	0,67583333	0,811
Diode chargeur	Nexperia	PMEG2020EPK	Farnell	2191802	5	Renaud	Requ	0,1325	0,6625	0,795
Self 22µh	Bourns	SRP1038A-220M	Farnell	2434023	1	Renaud	Requ	0,131666667	0,65833333	0,79
Buffer, non-inverseur	TI	SN741V125AD	Farnell	9592504	5	Renaud	Requ	0,250833333	1,25416667	1,505
Fusible 1,5A		0685F1500-01	Farnell	2834837	5	Renaud	Requ	0,065583333	0,65833333	0,787
Régulateur 3,3V	TI	LM3671MF-1.5/NOPB	Farnell	2492195	1	Renaud	Requ	0,214166667	0,6425	0,771
Condensateur 0,1µ	Multicomp	MC1206B104K201CT	Farnell	1855877	10	Renaud	Requ	0,128333333	0,64166667	0,77
Condensateur 18pF*4	Kemet	C1206C180J5GACTU	Farnell	1650895	10	Renaud	Requ	0,0975	0,975	1,17
Condensateur 2,2µ	Kemet	C1206C225K4RACTU	Farnell	9227911	5	Renaud	Requ	0,616666667	0,61666667	0,74
Self 2,2µ	Coilcraft	PFL3215-222MEB	Farnell	2288767	1	Renaud	Requ	0,052416667	0,52416667	0,629
M20	Harwin	M20-9993645	Farnell	1022263	1	Renaud	Requ	0,675833333	0,67583333	0,811
Condensateur 0,47µ	Multicomp	MCT1206R474KCT	Farnell	9406530	5	Renaud	Requ	0,050416667	0,50416667	0,605
Condensateur 0,01µ	Multicomp	MC1206B103K631CT	Farnell	1759518	5	Renaud	Requ	0,5	0,5	0,6
Condensateur 4,7µ	Multicomp	MC1206F475Z250CT	Farnell	1759452	5	Renaud	Requ	0,041666667	0,41666667	0,5
Condensateur 100n*4	WALSIN	1206B103J101CT	Farnell	2497065	10	Renaud	Requ	0,065583333	0,65833333	0,787
Fusible 0,500A		0685F0500-01	Farnell	2834833	3	Renaud	Requ	0,036666667	0,36666667	0,44
Led_temoins_program	Kingbright	KPTL-3216CGCK-01	Farnell	2217905	5	Renaud	Requ	0,128333333	0,64166667	0,77
Condensateur 10µ	WALSIN	1206F106Z100CT	Farnell	2497119	5	Renaud	Requ	0,125	0,625	0,75
Self 150µh	Panasonic	ELLCTP151MB	Farnell	2360397	1	Renaud	Requ	0,21	0,21	0,252
Résistance 2K		CRCW12062K00FKEA	Farnell	1469985	10	Renaud	Requ	0,198333333	0,19833333	0,0629
Condensateur 10µ	Kemet	C1206C106Z8VACTU	Farnell	9227903	5	Renaud	Requ	0,017083333	0,17083333	0,205
HB59 M59ce_620K_AO	Panasonic	ERJ8ENF6203V	Farnell	2307800	10	Renaud	Requ	0,016916667	0,16916667	0,203
Régulateur	ST	LM217T	Farnell	9756612	1	Renaud	Requ	0,016916667	0,16916667	6
Résistance 383K	Multicomp	MC0125W12061383K	Farnell	2142317	10	Renaud	Requ	0,016916667	0,16916667	0,203

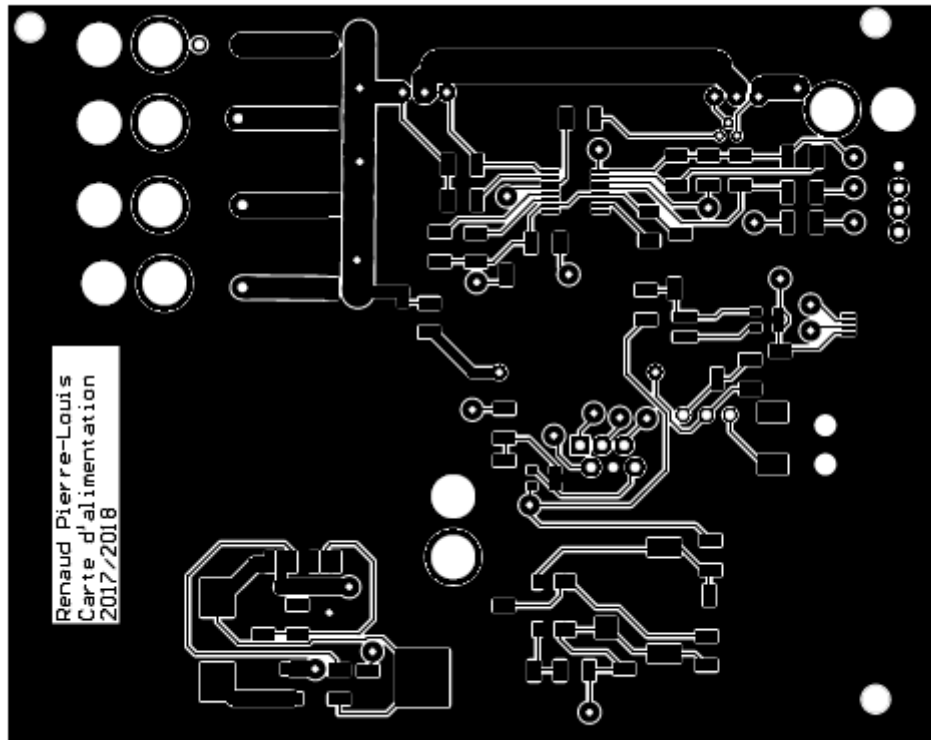
Régulateur	ST	LM217T	Farnell	9756612	1	Renaud	Reçu	0,016916667	0,16916667	6	
Résistance 383K	Multicomp	MC0125W12061383K	Farnell	2142317	10	Renaud	Reçu	0,016916667	0,16916667	0,203	
Oscillateur : 32,768M	Qantek	QTP732.76812B20R	Farnell	2508436	1	Renaud	Reçu	0,3825	0,3825	0,459	
Résistance 1,2ohm	Yageo	RC1206FR-071R2L	Farnell	9236538	10	Renaud	Reçu	0,015	0,15	0,44	
Résistance 1,2o 2W	Multicomp	MCKNP02SJO12JA10	Farnell	1903765	1	Renaud	Reçu	0,015	0,15	0,18	
Résistance 1M	TT	ASC1206-1M0FT5	Farnell	2079022	10	Renaud	Reçu	0,009333333	0,09333333	0,263	
Oscillateur :12M	Abracom	ABLS-12.000MHZ-B2-T	Farnell	1652551	1	Renaud	Reçu	0,21	0,21	0,252	
AOP	Microchip	MCP6001T-I/OT	Farnell	4974992	1	Renaud	Reçu	0,008916667	0,08916667	0,107	
Résistance 150	WELWYN	ASC1206-150RFT5	Farnell	2078996	10	Renaud	Reçu	0,017083333	0,17083333	0,205	
Résistance 4,7k *2	WELWYN	WCR1206-4K7FI	Farnell	1100209	10	Renaud	Reçu	0,016916667	0,16916667	0,203	
Résistance 47 *2	TT	WCR1206-47RFI	Farnell	1100156	10	Renaud	Reçu	0,016916667	0,16916667	0,203	
Résistance 75 *3	WELWYN	WCR1206-75RFI	Farnell	1100161	10	Renaud	Reçu	0,016916667	0,16916667	0,203	
Résistance 5,1K	TT	WCR1206-5K1FI	Farnell	1100211	10	Renaud	Reçu	0,091666667	0,45833333	0,09	
Résistance 51	TT	WCR1206-51RFI	Farnell	1100157	10	Renaud	Reçu	0,050416667	0,50416667	0,18	
Résistance 5,1K	TT	WCR1206-5K1FI	Farnell	1100211	10	Renaud	Reçu	1,608333333	1,60833333	1,93	
Résistance 1k	Multicomp	MCWR12X1001FTL	Farnell	2447473	10	Renaud	Reçu	4,941666667	4,94166667	0,0112	
Résistance 100	Multicomp	MCWR12X1000FTL	Farnell	2447454	10	Renaud	Reçu	0,008916667	0,08916667	0,107	
Résistance 1M *2	Multicomp	MCWR12X1004FTL	Farnell	2447478	10	Renaud	Reçu	0,008916667	0,08916667	0,107	
Résistance 33 *2	Multicomp	MC0125W1206533R	Farnell	9337318	10	Renaud	Reçu	0,00825	0,0825	0,099	
Résistance 1,5K	Multicomp	MCWR12X152 JTL	Farnell	2073893	10	Renaud	Reçu	0,008	0,08	0,096	
Résistance 100K	Multicomp	MC0125W12065100K	Farnell	9337024	10	Renaud	Reçu	0,007666667	0,07666667	0,092	
Résistance 19,1K	Panasonic	ERJ8ENF1912V	Farnell	2307489	10	Renaud	Reçu	0,05	0,48	0,0581	
Transistor_bkgd	Nexperia	BSV52	Farnell	1081314	5	Renaud	Reçu	0,091666667	0,45833333	0,55	
Resistance_620K_AOP	Panasonic	ERJ8ENF6203V	Farnell	2307800	10	Renaud	Reçu	0,05	0,05	0,605	
Connecteur batterie & moteur male		XT60	Hobbyking	601Ax5	1	Renaud	Reçu	1,61	1,6	1,93	
µcontrôleur	NXP	MC9S08QE128CLK	Mouser	841-MC9S08QE128CLK	1	Renaud	Reçu	4,941666667	4,94166667	5,93	
Led_bat	Kingbright	APFA3010SURKCGKSYKC	Mouser	304-APFA3010KCGKSYKC	1	Renaud	Reçu	0,616666667	0,61666667	0,74	
Diode(3,3v)	Littelfuse	TPSMB220A	Mouser	576-TPSMB220A	1	Renaud	Reçu	0,4	0,4	0,49	
Connecteur alim	RS pro		RS	476-157	1	Renaud	Reçu	3,32	3,31	3,98	
Régulateur de tension	Maxim integrated	MAX5033AASA+T	Mouser	700-MAX5033AASAT	1	Renaud	Reçu	3,096	3,096	2,58	
Chargeur	Ablic	S-8254AAWFT-TB-G	Mouser	628-S-8254AAWFT-TB-G	1	Renaud	Reçu	1,6	1,6	1,93	
Mosfet	Infineon	IPP80P03P4L04AKSA1	Mouser	726-IPP80P03P4L04AKS	2	Renaud	Reçu	1,29	2,58	1,55	
Condensateur	Taiyo yuden	EMK316BBJ476ML-T	Mouser	963-EMK316BBJ476ML-T	1	Renaud	Reçu	0,57	0,57	0,69	
Batterie	Turnigy		Hobbyking	9067000276-0	1	Renaud	Reçu	17,35833333	17,35833333	20,83	
BP_reset	TE-connectivity	1-1825910-5	RS	718-2376	5	Renaud	Reçu	0,11	0,55	0,66	
Prix total HT								78,45433333			
Prix total TCC								90,7642			

Nomenclature microcontrôleur

Nom du produit	Nom du fabricant	Référence fabricant	Nom du fournisseur	Référence fournisseur	Quantité	Prénom	etat commande	prix unitaire HT	total HT	prix TTC	commentaire
Buffer, non-inverseur	TI	SN74lv125AD	Farnell	9592204	5	Pilou		0,250833333	1,2541667	1,505	0,301
Oscillateur :12M	Abracom	ABLS-12.000MHZ-B2-T	Farnell	1652551	1	Pilou		0,21	0,21	0,252	0,252
Oscillateur : 32,768M	Qantek	QTP732.76812B20R	Farnell	2508436	1	Pilou		0,3825	0,3825	0,459	0,459
Microcontrôleur	NXP	MC9S08QE128CLK	Mouser	841-MC9S08QE128CLK	1	Pilou		4,941666667	4,9416667	5,93	5,93
Led_bat	Kingbright	APFA3010SURKCGKSYKC	Mouser	504-APFA3010KCGKSYKC	1	Pilou		0,616666667	0,6166667	0,74	0,74
Programmateurµi	NXP	MC9S08JS16CWI	Farnell	1704517	1	Pilou		2,816666667	2,8166667	3,38	3,38
Condensateur 10µ *2	WALSIN	1206F106Z100CT	Farnell	2497119	5	Pilou		0,125	0,625	0,75	0,15
Condensateur 100n *4	WALSIN	1206B103J101CT	Farnell	2497065	10	Pilou		0,065583333	0,6558333	0,787	0,0787
Condensateur 18pF * 4	Kemet	C1206C180J5GACTU	Farnell	1650895	10	Pilou		0,0975	0,975	1,17	0,117
Résistance 100K	Multicomp	MC0125W12065100K	Farnell	9337024	10	Pilou		0,007666667	0,0766667	0,092	0,0092
Résistance 47 *2	TT	WCR1206-47RFI	Farnell	1100156	10	Pilou		0,016916667	0,1691667	0,203	0,0203
Résistance 1,5K	Multicomp	MCMR12X152 JTL	Farnell	2073893	10	Pilou		0,008	0,08	0,096	0,0096
Résistance 1M *2	Multicomp	MCWR12X1004FTL	Farnell	2447478	10	Pilou		0,008916667	0,0891667	0,107	0,0107
Résistance 100	Multicomp	MCWR12X1000FTL	Farnell	2447454	10	Pilou		0,008916667	0,0891667	0,107	0,0107
Résistance 33 *2	Multicomp	MC0125W1206533R	Farnell	9337318	10	Pilou		0,00825	0,0825	0,099	0,0099
Résistance 150	WELWYN	ASC1206-150RFT5	Farnell	2078996	10	Pilou		0,017083333	0,1708333	0,205	0,0205
Résistance 75 *3	WELWYN	WCR1206-75RFI	Farnell	1100161	10	Pilou		0,016916667	0,1691667	0,203	0,0203
Résistance 4,7k *2	WELWYN	WCR1206-4K7FI	Farnell	1100209	10	Pilou		0,016916667	0,1691667	0,203	0,0203
Cavaliers	Pro-signal	PSG-JRBN40-FF	Farnell	2452751	1	Pilou		4,841666667	4,8416667	5,81	5,81
Led_temoin_program	Kingbright	KPTL-3216CGCK-01	Farnell	2217905	5	Pilou		0,128333333	0,6416667	0,77	0,154
BP_reset	TE-connectivity	1-1825910-5	RS	718-2376	5	Pilou		0,11	0,55	0,66	0,132
M20	Harwin	M20-9993645	Farnell	1022263	1	Pilou		0,675833333	0,6758333	0,811	0,811
Total HT									19,606667		
TOTAL TCC										23,3453	

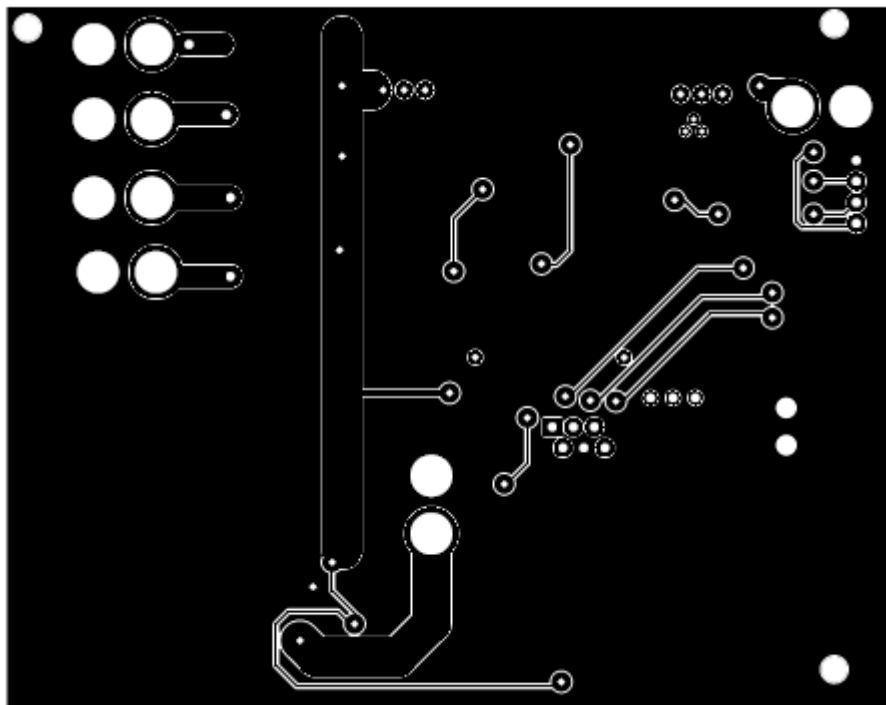
Partie alimentation

1. Top



Echelle 1 :1 Typon Top alimentation

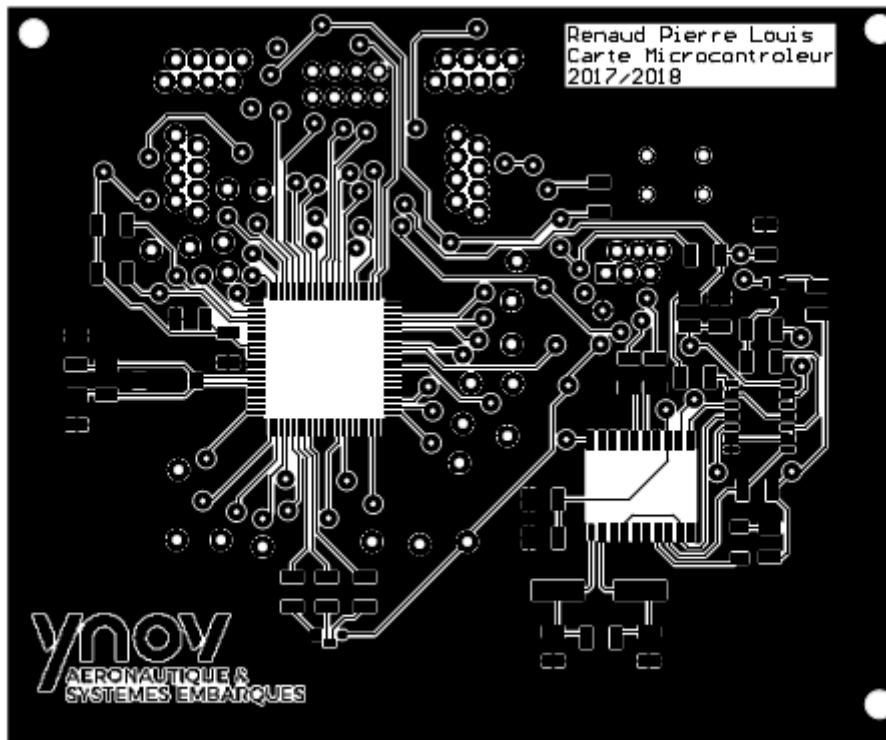
2. Bottom



Echelle 1 :1 Typon Bottom alimentation

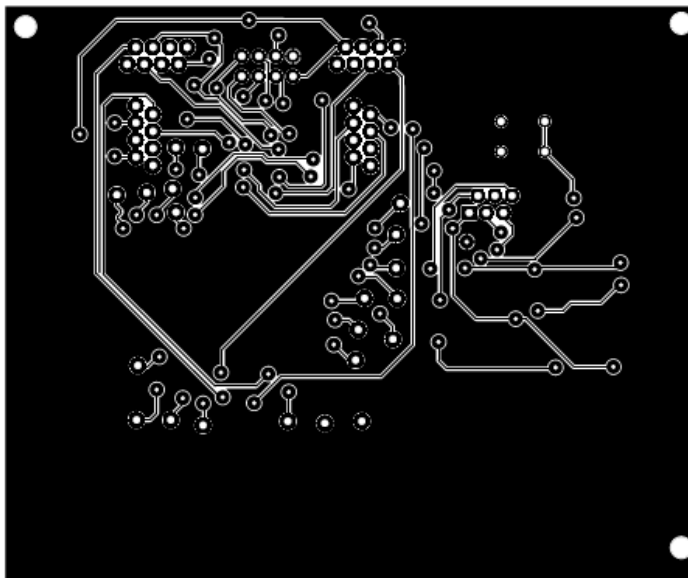
Partie microcontrôleur

1. Top



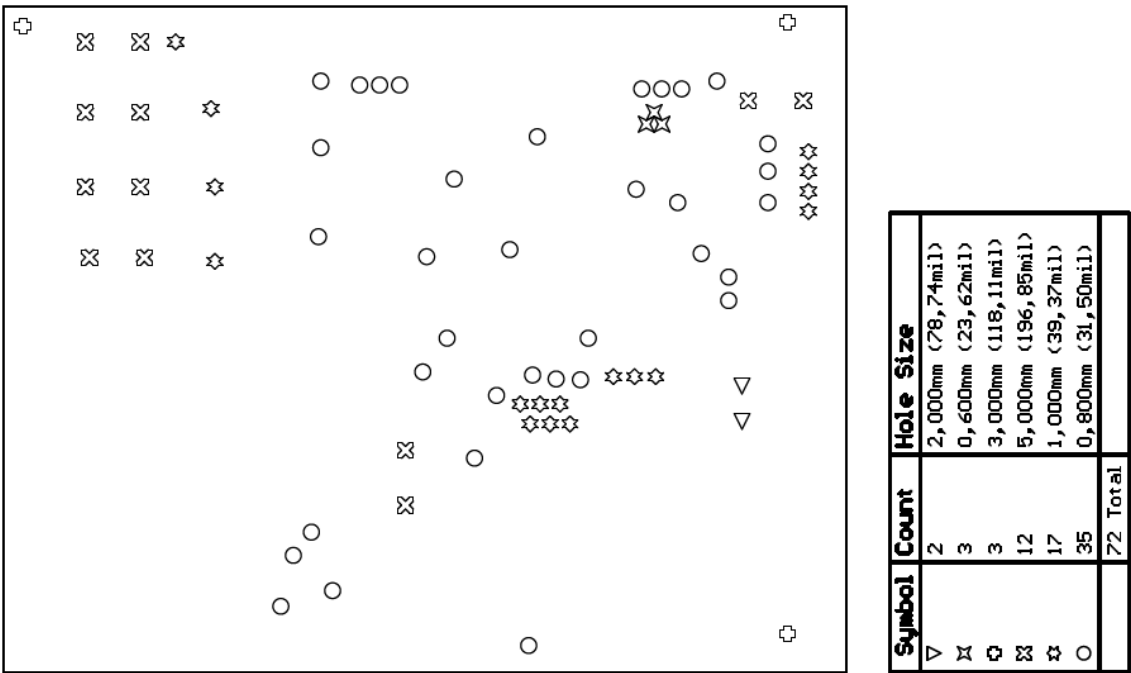
Echelle 1 :1 Typon Top microcontrôleur

2. Bottom



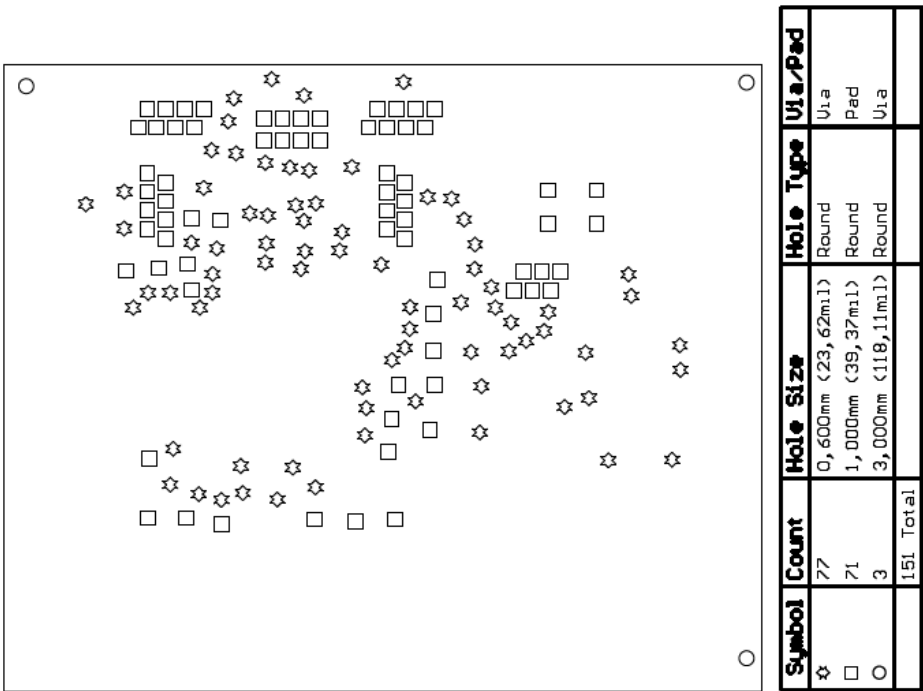
Echelle 1 :1 Typon Bottom microcontrôleur

Partie alimentation



Plan de perçage carte d'alimentation

Partie microcontrôleur



Plan de perçage carte microcontrôleur

Partie alimentation

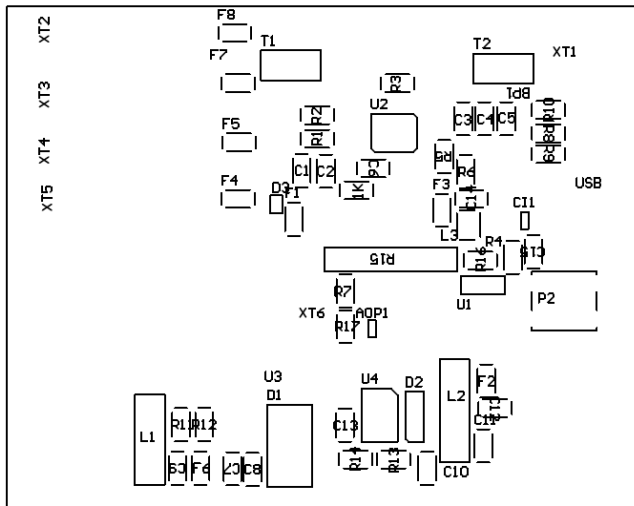


Schéma d'implantation des composants Top

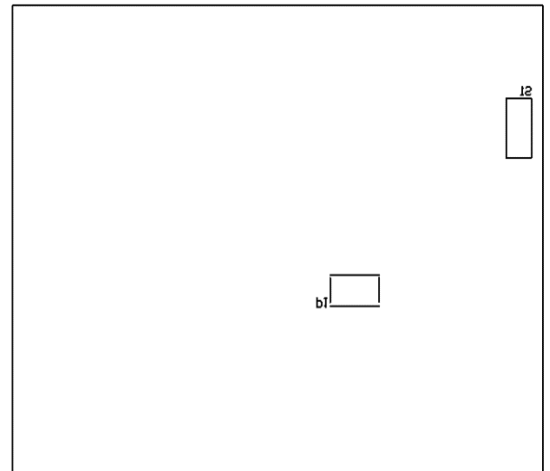


Schéma d'implantation des composants Bottom

Partie microcontrôleur

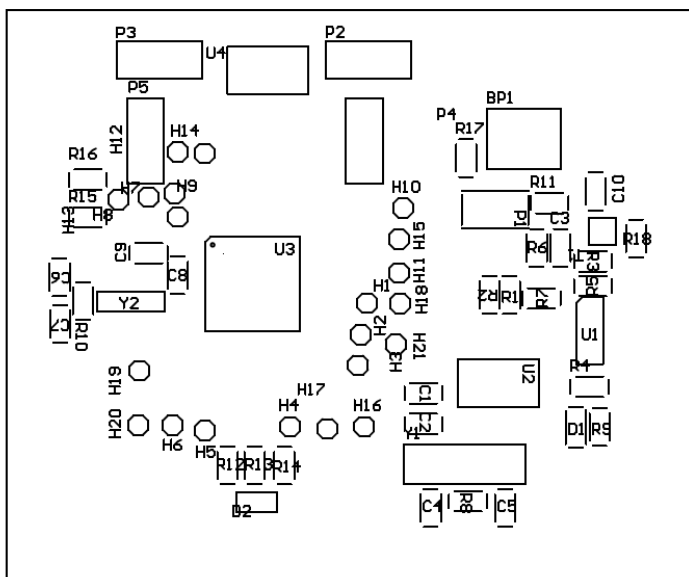
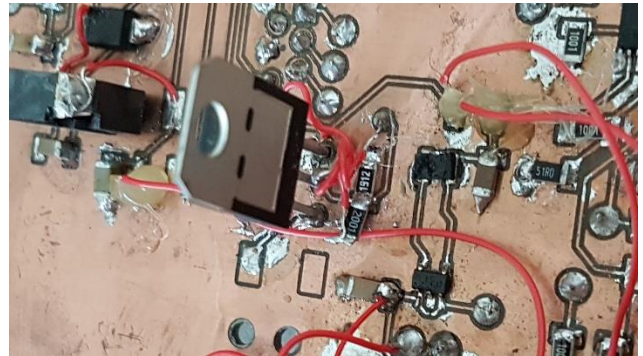


Schéma d'implantation des composants Top

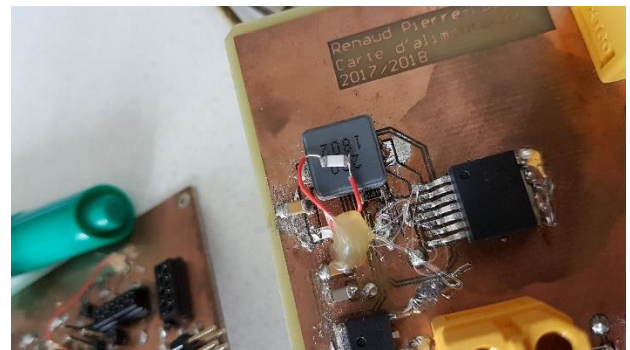
VII. Test et résolution des problèmes

La fabrication de la carte s'est faite par bloc d'alimentation pour faciliter le débogage. De multiples erreurs sur la schématique ont été repérées.

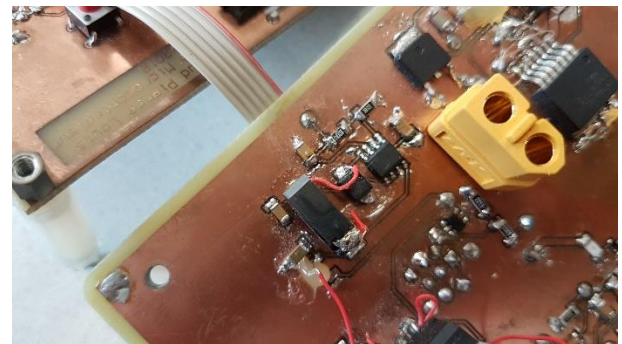
-Sur la partie chargeur de la batterie une erreur a été commise sur le pont diviseur, une résistance a été oubliée sur celui-ci. Pour corriger cette erreur les deux résistances ont été soudées l'une sur l'autre et une reprise filaire entre elles a été faite pour permettre le bon fonctionnement du chargeur.



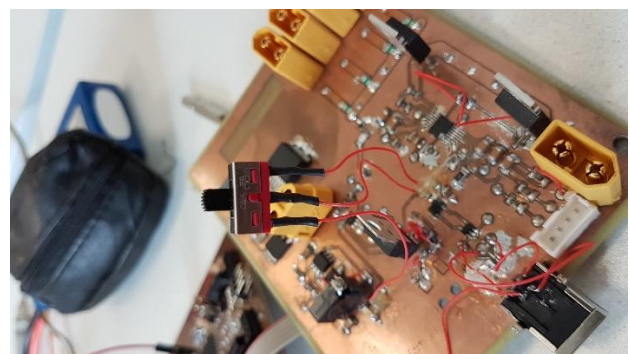
-Sur la partie alimentation 7.5V, une résistance mal positionnée a dû être déplacée pour le bon fonctionnement. De ce fait, deux raccords filaires pour connecter la résistance ont dû être faits.



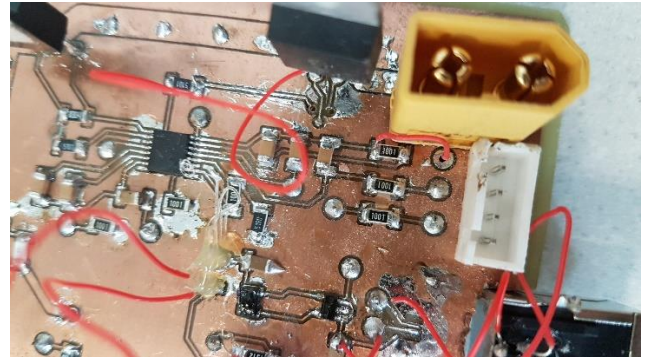
-Sur la partie 3.3V, j'ai sous-évalué la taille de la self blindée. En conséquence, j'ai dû la souder à la verticale pour régler le problème d'espace.



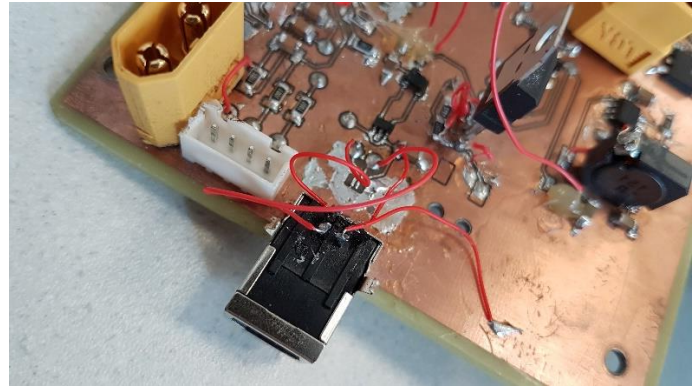
-un interrupteur trois positions a été placé par la suite pour permettre de switcher le régulateur 3.3V. En effet, l'un est connecté à la batterie et l'autre branché au port USB.



-Sur la partie gestion de la batterie le composant ne fonctionnait pas lorsqu'on l'alimentait. Ce problème est connu par le fabricant et il propose une solution. Il faut connecter la broche 2(VMP) à la broche 16(VDD). Egalement lors du routage un cheveu n'a pas été vu et une reprise filaire a été faite pour relier l'alimentation de la batterie au VDD du composant.

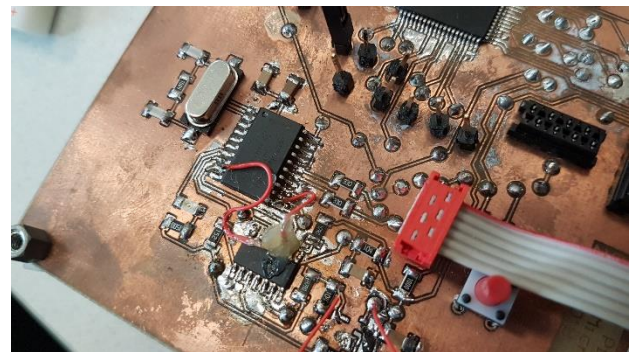


-Sur le connecteur USB : un connecteur micro USB a été initialement choisi et le problème est que celui-ci est très compliqué à souder. De ce fait, des pistes ont sauté en raison des différentes tentatives de soudage. Par la suite, un port USB type B a été installé mais des raccords filaires ont dû être mis en place. Des fils de la même taille ont été soudés pour les bornes D+ et D- pour avoir un minimum d'adaptation.

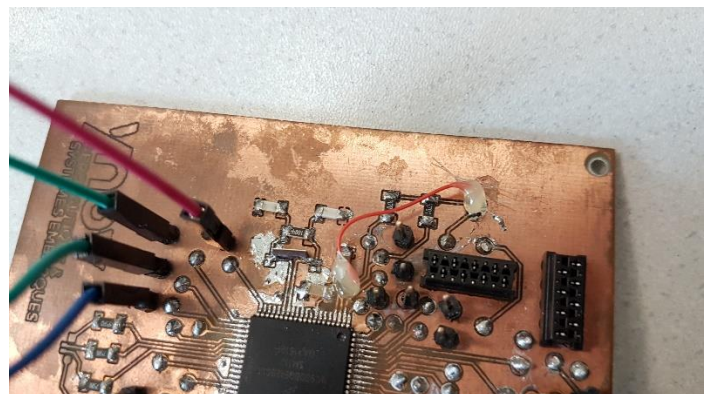


La carte microcontrôleur a elle été faite en une seule fois. Des erreurs ont été remarquées.

- Au niveau du programmeur : en raison de problèmes de soudure, des pads de connections sont partis. Pour corriger cela des reprises filaires ont été faites. Egalement pour des raisons inconnues le transistor reliant le buffer au microcontrôleur a cassé à plusieurs reprises. Du coup, pour faciliter les tests deux header avec un cavalier ont été installés.



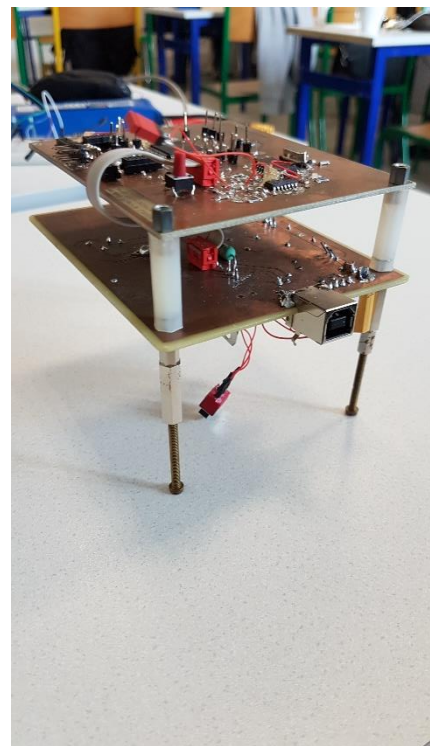
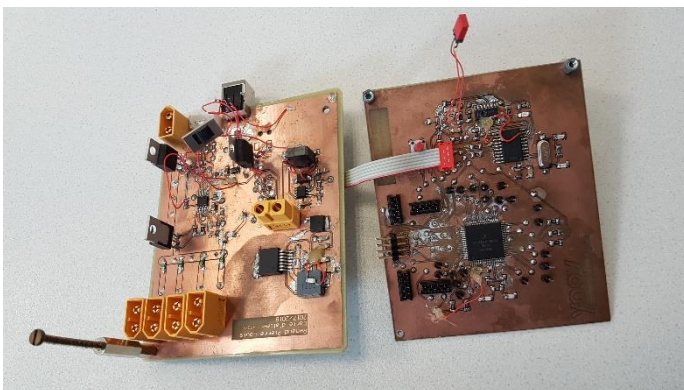
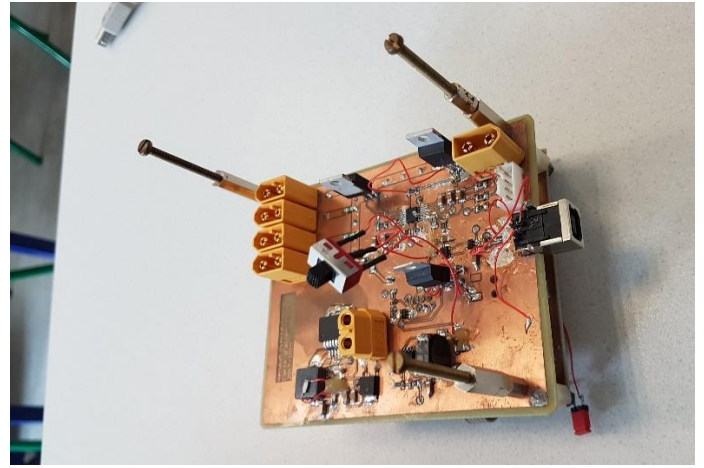
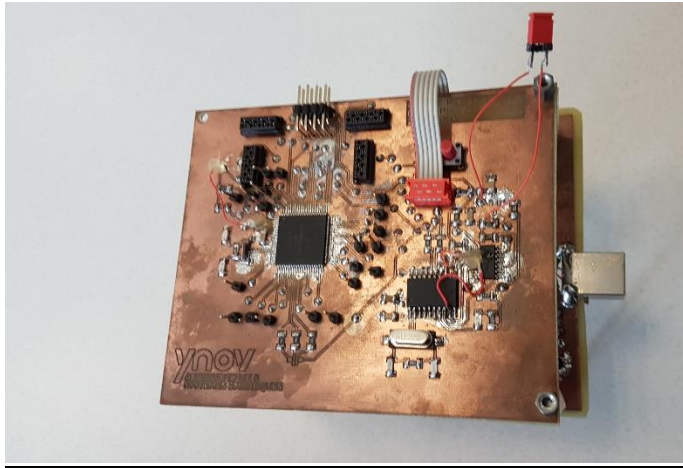
-Pour la partie microcontrôleur : au niveau de la schématique, le label connectant le 3.3V a été mal connecté. De ce fait, une reprise filaire pour relier le 3.3V au microcontrôleur a été réalisée.



NB : chaque reprise filaire a été renforcée avec de la colle pour éviter tout problème de court-circuit.

Malgré tous ces problèmes techniques les deux cartes fonctionnent correctement que ce soit indépendamment mais également lorsqu'elles sont rassemblées.

Voici un rendu de la carte :



Améliorations pour de prochaines cartes :

Pour la carte microcontrôleur, l'alimentation sera directement sur la carte et le connecteur USB sera également sur celle-ci pour avoir une meilleure adaptation des signaux.

VIII. Etude logicielle

Etudes de registres

Etude de la fonction Timer

Registre TPM1SC : \$48

TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
0	1	0	0	1	0	0	0

TOIE :

Active l'interruption du Timer.

RAZ : effacer TOIE

CLKSA : permet de sélectionner la source de la clock dans notre cas il s'agira du Bus rate.

Cette fonction Timer va permettre après réglage de déclencher une interruption tous les N temps. Dans cette interruption, on y place des instructions à exécuter.

Etude de la fonction CAN

Registres ADCSC1

7	6	5	4	3	2	1	0
CO	AI	AD	AD	AD	AD	AD	AD
CO	EN	CH	CH	CH	CH	CH	CH

ADCH [0 :4] : bits permettant la sélection de la voie de conversion. (00010)

ADCO : si à 1, le convertisseur réalise la conversion en continu de la voie présélectionnée.

COCO : indicateur de fin de conversion accessible en lecture seule. Si égal à 1 : la conversion est terminée. RAZ lors d'une écriture du registre ADCSC1 ou lecture du registre de résultat ADCRL.

Les autres bits sont laissés à leur état par défaut.

Registres ADCRL/H

Double registre de résultats en lecture seule.

7	6	5	4	3	2	1	0
AD	AD	AD	AD	AD	AD	AD	AD
CR	CR	CR	CR	CR	CR	CR	CR
H	H	H	H	H	H	H	H

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

AD CRL	AD CRL	AD CRL	AD CRL	AD CRL	AD CRL	AD CRL	AD CRL
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

Registre ADCCFG

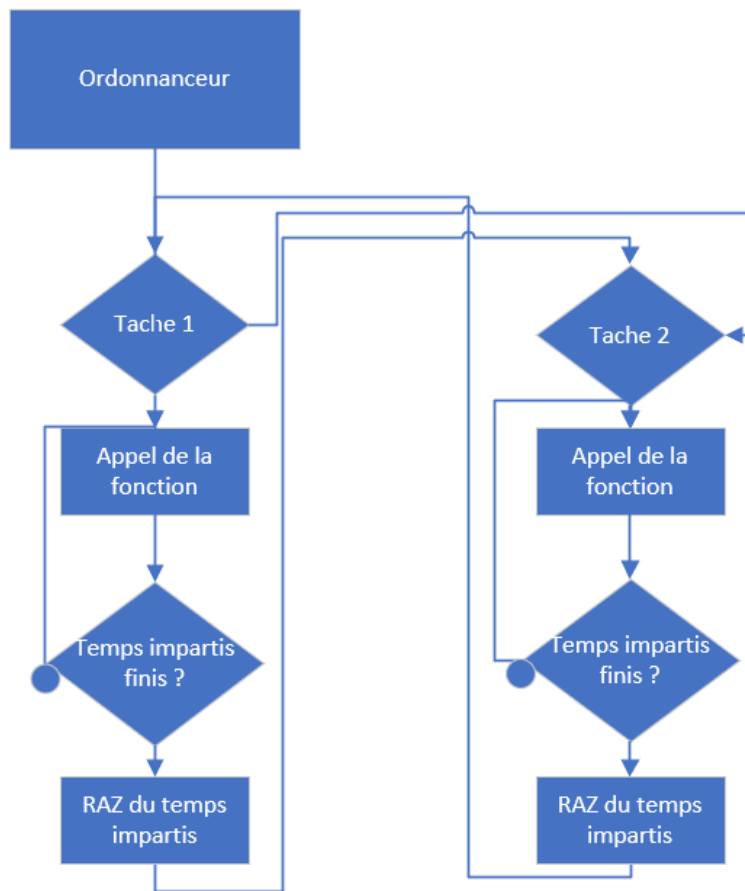
7	6	5	4	3	2	1	0
AD LP	A DI V	A DI V	ADL MSP	MO DE	MO DE	ADI CLK	ADI CLK

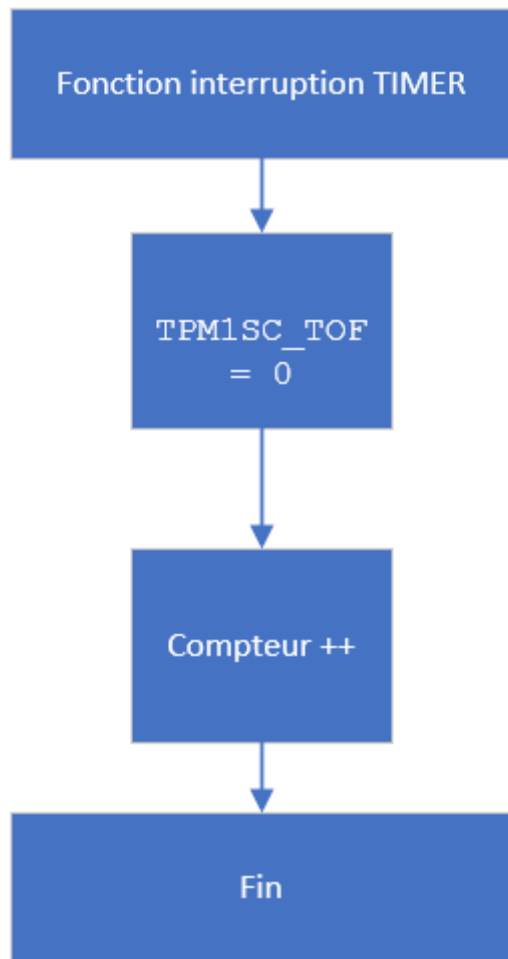
ADIV : définit la pré division de l'horloge de la fonction CAN

Mode : sélectionne la résolution du CAN. 00 résultat en 8 bits.

ADICLK : sélectionne la source de l'horloge du CAN. 00 sélectionne l'horloge du bus principal du microcontrôleur comme source.

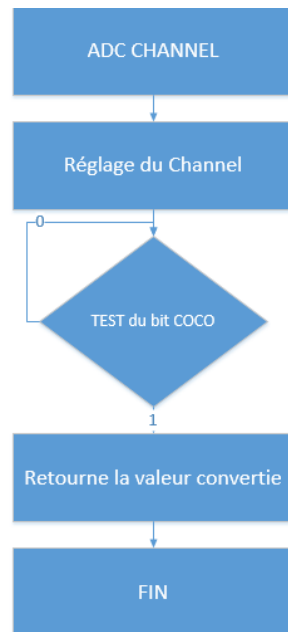
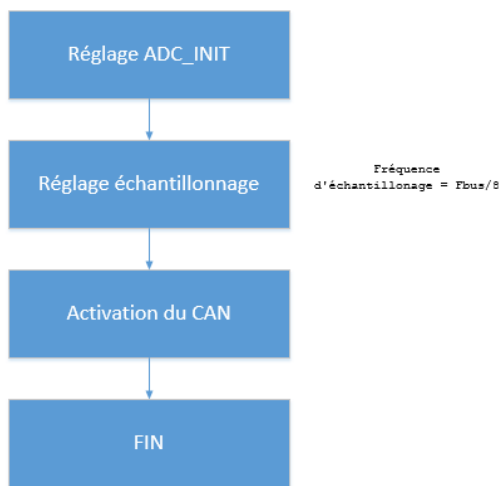
<p>Pas :</p> $\frac{\text{Tension pleine échelle}}{(2^n)-1} = \frac{3.3}{(2^8)-1} = 0.012$ <p>Avec :</p> <p>N= résolution du convertisseur</p> <p>Tension pleine échelle (3.3V)</p>	<p>Fréquence d'échantillonnage :</p> $\frac{F_{bus}}{8}$
---	--



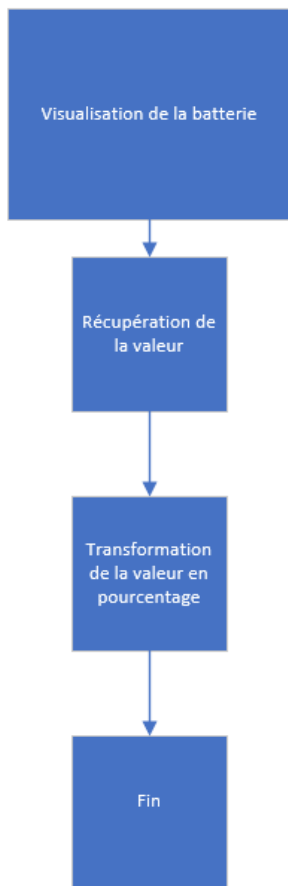


Ce programme permet d'appeler une fonction et de l'exécuter pendant N temps dans une boucle puis de passer à l'appel de fonction suivante.

Fonction ADC



Ce programme permet d'initialiser la fonction ADC et ensuite la convertir en une valeur sur cent. Cette technique permet d'avoir une assez bonne précision du pourcentage restant sur la batterie.



Par manque de temps la partie interface graphique utilisateur et le module radio n'ont pu être réalisés dans les délais impartis.

IX. Conclusion

En premier lieu, je tiens à remercier le personnel enseignant, Pierre Aubry, Aloïs Miclos et Steve Nguyen pour leur aide et les connaissances qu'ils nous ont apporté durant tout le projet. Je remercie également mes coéquipiers Clément Calliau, Clément Millereux et Vincent Villedieu pour leur contribution au projet.

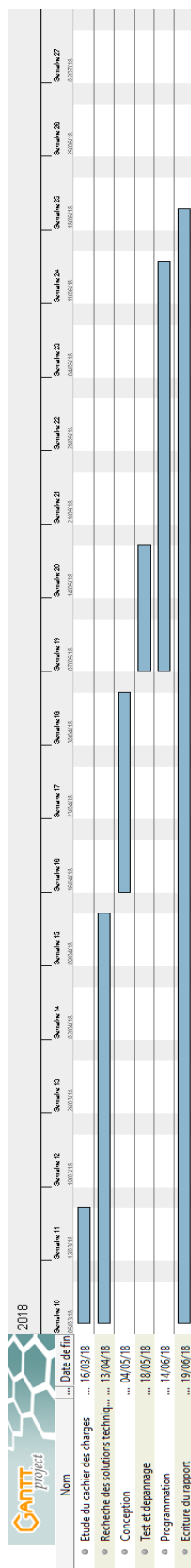
La réalisation technique de ce projet a fait appel à toute ma rigueur pour solutionner des erreurs sur la schématique. Celles-ci ont retardé mon avancée sur la réalisation des cartes d'alimentation et de microcontrôleur.

En conséquence, le temps m'a manqué pour terminer la partie transmission radio et l'interface utilisateur pour aboutir à un projet complet et fonctionnel.

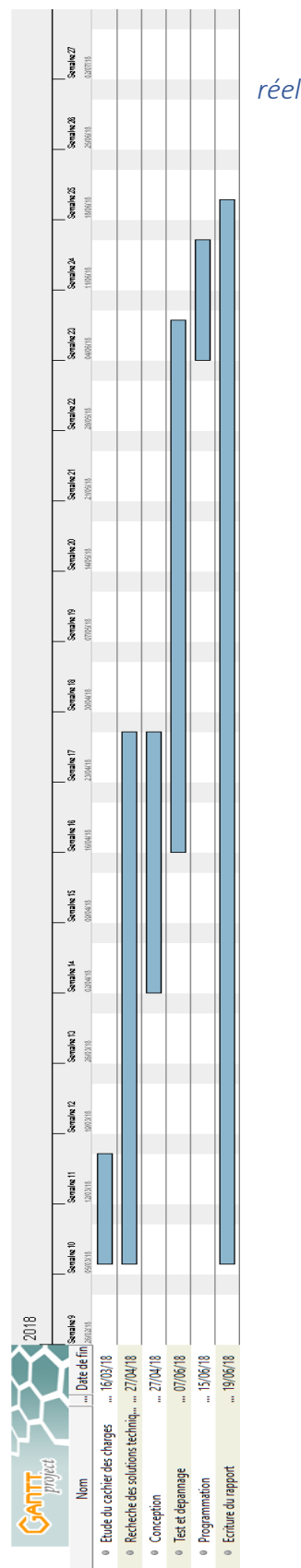
Cependant, le point positif est que ces deux cartes fonctionnent que ce soit de manière indépendante ou quand elles sont reliées. Elles permettent donc de remplir leur fonction dans la structure globale du projet de RobotCup.

Enfin, ce projet m'a permis d'avancer encore plus dans mes compétences en électronique et en programmation et me conforte un peu plus dans l'idée de travailler au terme de mes 5 ans d'école dans la robotique.

X. Annexes



Planning prévisionnel
Planning



Programme ADC

```
/* Including needed modules to compile this module/procedure */
#include "Cpu.h"
#include "Events.h"
#include "EInt1.h"
#include "can.h"
/* Include shared modules, which are used for whole project */
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"
```

```
#define bat 0x0B
unsigned char *toto = "55";
void ADC_init(void) {
    ADCCFG_ADICLK = 0x00; //fréquence d'échantillonnage
    ADCCFG_ADIV = 0x06; //fréquence d'échantillonnage = Fbus/8
    ADCSC1_ADCO = 0x01; //activation du ADC
    return;
}
char ADC(char channel) {
    ADCSC1_ADCH = channel;
    while (!ADCSC1_COCO)
        ;
    return ADCRL;
}
```

Init de l'ADC

```
#ifndef CAN_H_
#define CAN_H_
unsigned char* visu_bat(void);
char aADC(char);
void ADC_init(void);
```

```
#endif /* CAN_H_ */
```

ADC.h

```
void main(void) {
    char val_bat;

    PE_low_level_init();

    for (;;) {

        val_bat = ADC(bat); //récupère la valeur de l'adc
        val_bat = ((val_bat*100)/255); //convertie la valeur
    }

    /** Don't write any code pass this line, or it will be deleted during code generation. **/
    /** RTOS startup code. Macro PEX_RTOS_START is defined by the RTOS component. DON'T MODIFY THIS
#ifdef PEX_RTOS_START
    PEX_RTOS_START(); /* Startup of the selected RTOS. Macro is defined by the RTOS component. */
#endif
    /** End of RTOS startup code. **/
    /** Processor Expert end of main routine. DON'T MODIFY THIS CODE!!! **/
    for (;;) {
    }
    /** Processor Expert end of main routine. DON'T WRITE CODE BELOW!!! **/
} /** End of main routine. DO NOT MODIFY THIS TEXT!!! **/
```

Fonction appel ADC et conversion

```

⊕ **      Filename      : main.c
⊖ /*!
  ** @file main.c
  ** @version 01.12
  ** @brief
  **      Main module.
  **      This module contains user's application code.
  */
⊖ /*!
  ** @addtogroup main_module main module documentation
  ** @{
  */
/* MODULE main */

/* Including needed modules to compile this module/procedure */
#include "Cpu.h"
#include "Events.h"
#include "TPM1.h"
#include "PTE.h"
/* Include shared modules, which are used for whole project */
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"

/* User includes (#include below this line is not maintained by Processor Expert) */
static unsigned short compteur;
unsigned char tache = 1;
⊖ ISR(systick) {

    TPM1SC_TOF = 0;
    compteur++;
}

⊖ void main(void) {
    /* Write your local variable definition here */
    PTBDD = 0x32;
    /*** Processor Expert internal initialization. DON'T REMOVE THIS CODE!!! ***/
    PE_low_level_init();
    /*** End of Processor Expert internal initialization.          ***/

    switch (tache) {

    case 1:

        do {
            //Appel de la fonction.
            PTED_PTED5 = 0;
            PTED_PTED6 = 0;

        } while (compteur <= 50); // temps d'execution de la fonction

        do {

            //Appel de la fonction.
            PTED_PTED5 = 1;
            PTED_PTED6 = 1;

        } while (compteur <= 50); // temps d'execution de la fonction

        if (compteur >= 5) { //test de remise a zero du compteur

            compteur = 0;
        }
        tache++;

        break;
    }
}

```



```

case 2:
do {
    //Appel de la fonction.
    PTBD_PTBD5 = 0;
} while (compteur <= 200); // temps d'execution de la fonction
do {
    //Appel de la fonction.
    PTBD_PTBD5 = 1;
} while (compteur <= 200); // temps d'execution de la fonction
if (compteur >= 200) { //test de remise a zero du compteur
    compteur = 0;
}
do {
    //Appel de la fonction.
    PTBD_PTBD4 = 0;
} while (compteur <= 200); // temps d'execution de la fonction
do {
    //Appel de la fonction.
    PTBD_PTBD4 = 1;
} while (compteur <= 200); // temps d'execution de la fonction
if (compteur >= 200) { //test de remise a zero du compteur
    compteur = 0;
}

do {
    //Appel de la fonction.
    PTBD_PTBD1 = 0;
} while (compteur <= 200); // temps d'execution de la fonction
do {
    //Appel de la fonction.
    PTBD_PTBD1 = 1;
} while (compteur <= 200); // temps d'execution de la fonction

if (compteur >= 200) { //test de remise a zero du compteur

    compteur = 0;
}
tache++;
break;

break;
case 3:
    PTED_PTED5 = 0;
    PTED_PTED6 = 0;
    tache = 1;
    break;

default:
    break;

}

}

```

Régulateur de tension : LM3671

Tension de sortie 3.3V

Courant max : 0,600 A

Rendement : 95%

Calcul de puissance pour régulateur à découpage		
Vin		11,1
Vout		3,3
Iout		0,6
puissance		1,98
Rendement		0,84
Pc		0,37714286

Dissipation thermique :

Besoin d'un radiateur ?		
Temperature sans radiateur		102,492571 °C
Puissance Max		0,37714286 w
Rth j-a		165,7 °C/w
Temperature Ambiante		40 °C
Tjmax composant		125 °C
Pas de radiateur		

Régulateur de tension MAX5033

Tension de sortie : 3.3V

Courant max : 0.500A

Rendement : 94%

Calcul de puissance pour régulateur à découpage		
Vin		11,1
Vout		3,3
Iout		0,5
puissance		1,65
Rendement		0,94
Pc		0,10531915

Besoin d'un radiateur ?		
Temperature sans radiateur		57,9042553 °C
Puissance Max		0,10531915 w
Rth j-a		170 °C/w
Temperature Ambiante		40 °C
Tjmax composant		125 °C
Pas de radiateur		

Régulateur de tension LM2677

Tension de sortie : 7.5V

Courant max : 5A

Rendement : 92%

Calcul de puissance pour régulateur à découpage		
Vin		11,1
Vout		7,5
Iout		5
puissance		37,5
Rendement		0,92
Pc		3,26086957

Besoin d'un radiateur ?		
Temperature sans radiateur		154,130435 °C
Puissance Max		3,26086957 w
Rth j-a		35 °C/w
Temperature Ambiante		40 °C
Tjmax composant		125 °C
Radiateur nécessaire		

Régulateur de tension LM217

Tension de sortie : 13V

Courant max : 2.2A

Rendement : 84%

Calcul régulateur de tension

	Vin	15
	Vout	13
	Iout	1
	Pc	2

Besoin d'un radiateur ?

Temperature sans radiateur	110 °C
Puissance Max	2 w
Rth j-a	35 °C/w
Temperature Ambiante	40 °C
Tjmax composant	125 °C

Pas de radiateur